PROJETO E DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE E ACOMPANHAMENTO DE NOTIFICAÇÕES DE SPAM

Cristine Hoepers Klaus Steding-Jessen Marcelo H. P. C. Chaves
NIC BR Security Office – NBSO
Comitê Gestor da Internet no Brasil
{cristine,jessen,mhp}@nic.br

RESUMO

O envio de mensagens não solicitadas, conhecido como spam, é uma das formas de abuso na Internet que mais tem crescido nos últimos anos. Neste artigo são discutidos brevemente o histórico e a evolução do problema na Internet e o impacto que o volume de reclamações de spam gera no trabalho de grupos de resposta a incidentes e abusos na Internet. Posteriormente são discutidos o projeto e a implementação de um sistema para controle e acompanhamento de notificações de spam, bem como os resultados obtidos após sua implantação em um grupo de resposta a incidentes.

ABSTRACT

The use of the Internet to send unsolicited email, also known as spam, has dramatically increased in the past few years. In this paper we briefly discuss the history and the evolution of the problem, and the impact spam complaints have on computer security incident response teams' day-to-day operations. Then we present a tool developed to control and keep track of spam notifications, as well as some results obtained after the tool has been adopted by an incident response team.

1 INTRODUÇÃO

O termo *spam* é utilizado para descrever o envio indiscriminado de mensagens não solicitadas ou com conteúdo inapropriado, especialmente em grandes quantidades e com teor comercial [1, 2].

A popularização do termo *spam* para denotar o envio em massa de *emails* não solicitados remonta a um incidente ocorrido em abril de 1994, em que dois advogados enviaram uma mensagem não solicitada, simultaneamente, para seis mil grupos de *Usenet News* [3, 4].

Desde então, o *spam* é uma das formas de abuso da Internet que mais tem crescido [4], sendo atualmente a origem de boa parte do volume de *emails* e afetando o tráfego na rede. Devido à proporção que a questão do *spam* tomou, várias abordagens técnicas têm sido utilizadas para lidar com o problema, como por exemplo, implantação de recomendações para configuração de sistemas de *email* [5, 6], uso de listas de bloqueio e de mecanimos de filtragem baseados em características estatísticas de mensagens [7].

Em fevereiro de 2003 foi criado o *Anti-Spam Research Group*[1] (ASRG), ligado à *Internet Research Task Force* (IRTF), com o objetivo de entender o problema do *spam* e coletivamente propor e avaliar soluções [8].

Complementarmente às iniciativas mencionadas, existem algumas entidades que se dedicam a receber reclamações de *spam* e redirecioná-las para os responsáveis pelas redes envolvidas com a atividade abusiva.

Neste artigo é apresentado um sistema para controle e acompanhamento de reclamações de *spam* recebidas destas entidades e os resultados obtidos após sua implantação em um grupo de resposta a incidentes.

Este artigo é organizado como segue. Na seção 2 é discutida a motivação para o desenvolvimento do sistema e para a processamento de reclamações relativas ao envio de *spam*. Na seção 3 é discutida a visão geral da arquitetura do sistema para controle e acompanhamento de notificações de *spam* e na seção 4 é descrita a implementação dos diversos módulos que o compõem. Os resultados observados são apresentados na seção 5, as propostas para trabalhos futuros e as conclusões são discutidas nas seções 6 e 7, respectivamente.

2 MOTIVAÇÃO

Grupos de segurança, tratamento de abusos e resposta a incidentes recebem diariamente um grande volume de reclamações relativas a abusos originados ou destinados a suas redes. Devido a este grande volume é necessário um processo de triagem que normalmente prioriza o atendimento a incidentes envolvendo ataques e comprometimentos de sistemas e redes, e atividades que infrinjam as políticas de segurança de suas instituições [9].

Reclamações relativas a *spam*, mesmo não sendo diretamente relacionadas com incidentes considerados emergenciais, contém muitas informações relevantes sobre sistemas que podem estar com problemas de configuração ou sendo abusados por terceiros.

A entidade SpamCop[2], por exemplo, possui um serviço que processa reclamações de *spam* e as envia para os responsáveis pelas redes envolvidas com o abuso. Estas reclamações envolvem máquinas com *proxies* ou *relays* abertos, possivelmente sendo abusados, ou máquinas hospedando páginas com informações de produtos e serviços sendo oferecidos no *spam* (*spamvertised website*).

[1] http://www.irtf.org/asrg/

[2] http://www.spamcop.net/

Um *proxy* [10] é um serviço que atua como intermediário entre um cliente e um servidor, e um *relay* [11] é uma funcionalidade do serviço SMTP que permite receber *emails* de clientes e retransmití-los, sem modificações, para outro servidor SMTP. Máquinas com *proxies* e *relays* abertos podem ser utilizadas indiscriminadamente por terceiros para enviar *spam*, dificultando a identificação da real origem. *Proxies* abertos podem também ser usados como pontes para realização de invasões e desfigurações de páginas *Web* ou como meio de obter anonimato ao cometer crimes como estelionato ou pornografia envolvendo crianças [12].

Ao receber informações sobre máquinas mal configuradas e possivelmente sendo abusadas em sua rede, um administrador ou analista de segurança pode rapidamente solucionar o problema e evitar que sua rede seja utilizada indevidamente por terceiros.

O grupo de resposta a incidentes onde o sistema aqui apresentado foi desenvolvido recebe diariamente cerca de 8.500 *emails* em seu endereço destinado a reclamações de *spam*. Destes *emails*, cerca de 90% são reclamações enviadas pelo SpamCop, como pode-se ver na Tabela 1.

Tabela 1: Total de mensagens recebidas no primeiro semestre de 2003 pelo grupo de resposta a abusos, mensagens enviadas pelo SpamCop e respectiva porcentagem.

Mês	Mensagens	SpamCop	%
jan	163.213	141.805	86.88
fev	203.897	179.332	87.95
mar	194.278	167.151	86.04
abr	326.916	292.583	89.50
mai	364.668	332.430	91.16
jun	297.504	270.314	90.86
Total	1.550.476	1.383.615	89.24

Das reclamações enviadas pelo SpamCop, aquelas relativas a *proxies* e *relays* abertos e *spamvertised website* possuem um formato bem definido e constituem a maioria absoluta, como pode ser visto na Tabela 2.

Tabela 2: Reclamações dos tipos *spamvertised website*, *proxy* aberto e *relay* aberto, recebidas do SpamCop.

Mês	*Spamv.*	*Proxy*	*Relay*	Outros
jan	53.843	31.670	1.127	55.165
fev	88.732	47.058	712	42.822
mar	65.511	60.837	779	40.021
abr	189.851	56.050	225	46.448
mai	216.824	59.981	471	55.138
jun	152.521	56.197	583	60.991
Total	767.282	311.793	3.897	300.585

Considerando as características citadas acima, decidiu-se desenvolver um sistema que inicialmente fará o processamento efetivo apenas das mensagens recebidas da instituição SpamCop e dos seguintes tipos: *proxy* aberto, *relay* aberto e *spamvertised website*.

Este sistema foi projetado para desempenhar as seguintes funções:

- processar reclamações de *spam*, identificando aquelas que são provenientes do SpamCop e dos tipos sendo tratados;
- agrupar as reclamações do SpamCop por responsáveis pelas redes e tipo de abuso;
- gerar notificações para os responsáveis pelas redes e máquinas presentes nas reclamações do SpamCop;
- gerar estatísticas relativas ao tipo de abuso notificado e manter um histórico relativo a cada rede;
- gerar estatísticas relativas às redes sendo abusadas para envio de *spam*, considerando número de endereços IP listados, reincidências, etc;
- gerar estatísticas gerais sobre todas as reclamações de *spam* recebidas.

3 ARQUITETURA DO SISTEMA

O sistema em questão lida essencialmente com arquivos contendo mensagens de reclamação de *spam* (*mailboxes*) e sua arquitetura é composta por dois módulos: acompanhamento de notificações e geração de estatísticas. Estes são subdivididos em módulos menores, como pode ser visto na Figura 1, e serão descritos posteriormente.

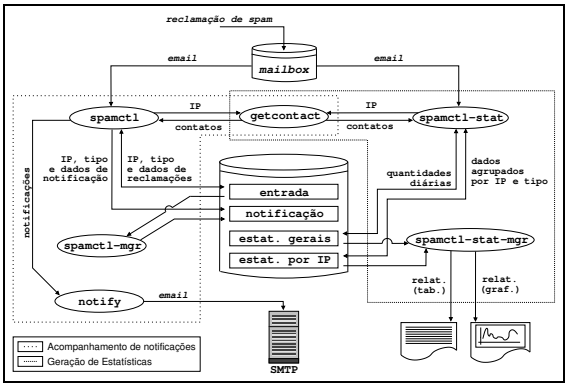

Figura 1: Interação e fluxo de dados entre módulos e meios de armazenamento.

A entrada de dados destes dois módulos consiste em um arquivo contendo todos os *emails* de reclamação de *spam* recebidos. As saídas do sistema são:

módulo de acompanhamento de notificações: mensagens de notificação, contendo informações agrupadas, sobre abusos provenientes de diversas redes com

um mesmo responsável.

módulo de geração de estatísticas: relatórios com informações sumarizadas para todos os tipos de reclamações e sobre as redes com maior número de reclamações registradas, agrupadas por responsável ou por bloco CIDR[3].

Seguem descrições das funcionalidades dos grandes módulos integrantes do sistema.

3.1 Acompanhamento de Notificações

Este módulo agrupa dados de reclamações e envia mensagens de notificação contendo tais dados para os responsáveis pelas redes envolvidas.

As reclamações processadas são usualmente recebidas em intervalos de tempo curtos, contendo informações relativas a um mesmo endereço IP e a um mesmo tipo ou relativas a endereços IP diferentes, mas com um mesmo responsável de contato.

Dada esta característica optou-se por agrupar reclamações, referentes a redes administradas por um mesmo responsável e com o mesmo tipo de reclamação, respeitando um certo período pré-determinado de tempo.

Deste modo, cada notificação enviada pode possuir um relatório consolidado com os dados referentes a diversas reclamações recebidas dentro do período de tempo pré-determinado, que corresponde ao período de permanência máximo no sistema.

Seguem breves descrições dos módulos menores que constituem o módulo de acompanhamento de notificações.

`spamctl`: responsável por agrupar, em um arquivo de entrada, dados de reclamações de *spam* que envolvam o mesmo endereço IP e o mesmo tipo. Quando o período de permanência máximo destes dados no arquivo de entrada é alcançado, eles são movidos para o arquivo de notificação. Este arquivo é processado para gerar a saída deste módulo, que é composta pelos dados das reclamações, agrupados por endereço IP e tipo, acompanhados dos dados de contato, obtidos através do módulo `getcontact`.

`getcontact`: retorna uma lista de contatos responsáveis por um dado endereço IP e, opcionalmente, o bloco CIDR ao qual este endereço pertence. Estes dados são obtidos através de consultas a servidores de WHOIS[4].

`notify`: responsável pela geração e envio de notificações por *email*, a partir das mensagens fornecidas pelo módulo `spamctl`. Permite que diversas notificações contendo o mesmo responsável e o mesmo tipo de reclamação sejam agrupadas antes do envio.

`spamctl-mgr`: utilizado para mover dados do arquivo de entrada para o arquivo de notificação, em casos que fujam à regra geral do período de permanência, citado na descrição do módulo `spamctl`.

3.2 Geração de Estatísticas

Este módulo é responsável por contabilizar e armazenar informações utilizadas na geração de estatísticas sobre as reclamações de *spam*.

Seguem breves descrições dos módulos menores que constituem o módulo de geração de estatísticas.

`spamctl-stat`: responsável por identificar e extrair das reclamações de *spam* as informações a serem armazenadas em dois arquivos que serão utilizados pelo módulo `spamctl-stat-mgr`. O primeiro arquivo, utilizado para estatísticas gerais, armazena para todas as reclamações a quantidade diária de cada uma das categorias. O segundo arquivo armazena dados das reclamações efetivamente tratadas por este sistema, agrupadas pelo par endereço IP e tipo de reclamação, acompanhadas das informações de contato e de bloco CIDR obtidas pelo módulo `getcontact`. Este arquivo é utilizado na geração das estatísticas por endereço IP.

`spamctl-stat-mgr`: responsável por emitir relatórios sobre as reclamações de *spam*, com base nos dados armazenados nos arquivos de estatísticas gerados pelo módulo `spamctl-stat`. Tais relatórios podem conter informações sobre:

- número de reclamações de diversas categorias, com valores diários e totais, extraídas do arquivo de estatísticas gerais;
- número de reclamações específicas por endereço IP, responsável ou bloco CIDR, extraídas do arquivo de estatísticas por IP;
- redes que possuem o maior número de reclamações associadas, em ordem decrescente, e agrupadas por responsáveis ou CIDRs.

4 IMPLEMENTAÇÃO

Como a função básica do sistema aqui apresentado é processar reclamações de *spam*, que são mensagens de texto, fez-se necessário escolher uma linguagem de programação que facilite esse processamento e que permita extrair, de forma simples, partes relevantes de tais mensagens.

Portanto, o sistema foi desenvolvido na liguagem Perl por ser esta uma linguagem fortemente indicada para o processamento de textos, de fácil compreensão e com alto grau de portabilidade.

[3] `http://www.ietf.org/rfc/rfc1518.txt`
[4] `http://www.ietf.org/rfc/rfc954.txt`

As subseções seguintes apresentam as estruturas de dados utilizadas pelo sistema, detalhes da implementação de cada um de seus módulos e uma breve discussão sobre métodos de programação segura levados em consideração durante o seu desenvolvimento.

4.1 Estruturas de Dados

As informações extraídas das reclamações são armazenadas em tabelas *hash* [13] que, em Perl, são estruturas intrínsecas da linguagem [14]. Estas estruturas podem ser armazenadas tanto em memória como em disco.

Convencionou-se neste artigo que o termo registro denota um par chave/valor de uma tabela *hash* e que o termo arquivo refere-se a um *hash* armazenado em disco.

Seguem as descrições dos dados contidos em cada um dos registros manipulados pelos módulos que compõem o sistema.

registro de entrada: a chave deste registro é composta pelo endereço IP e pelo tipo de reclamação. O valor associado a cada chave é composto pelos seguintes dados: data de criação do registro (*timestamp*), contatos responsáveis pelo endereço IP, lista das datas de recebimento das reclamações associadas, menor e maior data nesta lista, e textos contendo as partes relevantes de cada reclamação[5]. Este tipo de registro é armazenado no arquivo de entrada.

registro de notificação: contém partes dos dados de um registro de entrada. Sua chave é composta pelo endereço IP, pelo tipo de reclamação e pela data em que foi criado (*timestamp*). O valor associado a cada chave é composto pelos seguintes dados: contatos responsáveis pelo endereço IP e textos contendo as partes relevantes de cada reclamação, como descrito para o registro de entrada. Este tipo de registro é armazenado no arquivo de notificação.

registro de estatísticas gerais: as chaves destes registros são as datas em que houve recebimento de reclamações. O valor associado a cada chave corresponde à contagem diária de reclamações. Ele é composto pelo total de reclamações e por subtotais para cada uma das categorias. Este tipo de registro é armazenado no arquivo de estatísticas gerais.

registro de estatísticas por IP: a chave de cada registro é composta pelo endereço IP e pelo tipo de reclamação. O valor associado a cada chave é composto pelos seguintes dados: contatos responsáveis pelo endereço IP, bloco CIDR ao qual o IP pertence, e lista das datas de recebimento das reclamações associadas, em ordem crescente. Este tipo de registro é armazenado no arquivo de estatísticas por IP.

[5] As partes relevantes de uma reclamação contém o tipo, o endereço IP envolvido, a data de envio e a página *Web* no *site* do SpamCop que referencia a reclamação.

4.2 Módulos

Antes de discutir a implementação, faz-se nescessário apresentar algumas considerações relacionadas à execução dos módulos `spamctl` e `spamctl-stat`. Estas considerações são:

- Antes de iniciar o processamento das reclamações, são feitas cópias de segurança dos arquivos manipulados pelos módulos citados acima. A idéia é possibilitar a recuperação destes arquivos, em um estado consistente, caso ocorra algum problema durante seus processamentos;
- Por questões de desempenho, os registros armazenados em arquivos são carregados para tabelas *hash* em memória e, somente depois de processadas todas as reclamações, os registros atualizados em memória são descarregados para os respectivos arquivos;
- Alguns procedimentos dos módulos acima citados verificam se o IP extraído de cada reclamação pertence a alguma rede brasileira, consultando uma lista de blocos CIDR alocados para o Brasil[6].

Seguem as descrições de alguns detalhes de implementação dos módulos que constituem este sistema.

4.2.1 spamctl

O módulo `spamctl` recebe como parâmetros o período de permanência máximo dos dados de reclamações nos registros de entrada e os *mailboxes* que deverão ser processados. Para cada reclamação de *spam* em um *mailbox* especificado, o módulo verifica se ela foi enviada pelo SpamCop e se é de algum tipo efetivemente tratado. Em caso afirmativo, executa os seguintes passos:

1. Extrai o endereço IP envolvido e verifica se pertence a alguma rede brasileira;
2. Extrai a data de recebimento e o texto contendo a parte relevante da reclamação;
3. Busca pelo registro de entrada associado por um mesmo IP e tipo. Se não encontrar, obtém os contatos responsáveis pelo IP, cria um novo registro e o preenche com os dados obtidos neste passo;

[6] Esta lista é gerada a partir de arquivos fornecidos mensalmente pelo ARIN (`ftp://ftp.arin.net/pub/stats/arin`) e LACNIC (`ftp://ftp.lacnic.net/pub/stats/lacnic`).

4. Acrescenta a data de recebimento à lista de datas associadas. Atualiza a menor data ou a maior data, comparando-as com a data de recebimento;

5. Acrescenta o texto contendo a parte relevante da reclamação aos outros já agrupados (se existirem);

6. Verifica se a diferença entre a maior e a menor data excede o período de permanência máximo estipulado. Em caso positivo, move os dados do registro de entrada para um novo registro de notificação.

A saída do `spamctl` é gerada após o processamento de todas as reclamações, com base nos registros de notificação, e direcionada para um arquivo no formato ASCII. Este arquivo é posteriormente utilizado como entrada para o módulo `notify` (seção 4.2.3). A Figura 2 apresenta um exemplo dessa saída.

```
To: contato@dominio-1.example.net
Subject: <192.0.2.1> - host(s) listado(s) como Open Proxy
---begin logs---
http://spamcop.net/w3m?i=z317681570zbbd5aabe9ea8df0f191699cc00e09b2az
Email from 192.0.2.1 / Mon, 30 Jun 2003 19:23:28 +0200 (MEST)
> 192.0.2.1 is an open proxy, more information:
> http://spamcop.net/mky-proxies.html
...
http://spamcop.net/w3m?i=z323226007z75a49a5159310205406c1e915807d2f7z
Email from 192.0.2.1 / Sat, 5 Jul 2003 01:05:27 +1000
> 192.0.2.1 is an open proxy, more information:
> http://spamcop.net/mky-proxies.html
---end logs---
To: contato@dominio-2.example.net
Subject: <192.0.2.2> - host(s) listado(s) como Open Relay
---begin logs---
http://spamcop.net/w3m?i=z317819502z2445e6397f8b011e4b16a89d0d1845e5z
Email from 192.0.2.2 / 30 Jun 2003 18:43:28 -0000
> 192.0.2.2 is an open relay, more information:
> http://spamcop.net/mky-relay.html
...
http://spamcop.net/w3m?i=z323168673za2ddeb8f0f87efd639f023e528c6e186z
Email from 192.0.2.2 / Fri, 4 Jul 2003 20:40:46 -0300
> 192.0.2.2 is an open relay, more information:
> http://spamcop.net/mky-relay.html
---end logs---
```

Figura 2: Saída gerada pelo módulo `spamctl`.

4.2.2 getcontact

O módulo `getcontact` faz consultas a servidores WHOIS utilizando o programa `jwhois`[7], devido às seguintes funcionalidades:

- realização de consultas sempre para o servidor mais específico, através de um arquivo de configuração extensível;
- possibilidade de redirecionamento da consulta para outro servidor, em função da resposta obtida;
- manutenção de um *cache* das informações obtidas, por endereço IP.

[7] `http://www.gnu.org/software/jwhois/`

Os *emails* de contato são obtidos através do processamento das consultas ao servidor WHOIS. Por questões de eficiência é mantido um um *cache* das informações por bloco CIDR, de modo que próximas consultas a endereços IP contidos nesse bloco estarão disponíveis de imediato, sem necessidade de uma consulta externa.

Os *emails* obtidos nas consultas são então comparados com uma tabela de conversão, que permite que determinados endereços possam ser inseridos ou complementados. Isso é particularmente útil para informações de contato incorretas ou incompletas.

4.2.3 notify

O módulo `notify` recebe como entrada um arquivo com blocos de texto contendo campos de contato, *subject* e *logs*. Um exemplo desse tipo de arquivo de entrada é mostrado na Figura 2.

Os blocos que possuem as mesmas informações de contato e mesma categoria de *subject* são agrupados[8] e separados por delimitadores, para facilitar a leitura.

Na versão agrupada, os componentes variáveis do campo *subject*, como endereços IP, são substituídos por um contador que indica o número de diferentes endereços IP contidos nos *logs*.

O módulo permite a escolha de um texto padrão, inserido no início de cada notificação, além da escolha do conteúdo de alguns *headers* de *email*, tais como ‘`Cc:`’, ‘`Reply-To:`’, ‘`From:`’, etc.

4.2.4 spamctl-mgr

O módulo `spamctl-mgr` foi implementado para tratar de casos específicos, onde a diferença entre a maior e a menor data de recebimento nos registros de entrada não alcança o período de permanência máximo utilizado na execução do módulo `spamctl` (seção 4.2.1).

O módulo aplica um procedimento que verifica se a diferença entre a data atual e a data de criação de cada registro de entrada é maior ou igual a um determinado número de dias, fornecido como parâmetro. Em caso positivo, os dados deste registro são movidos para um novo registro no arquivo de notificação.

4.2.5 spamctl-stat

O módulo `spamctl-stat` recebe como parâmetro os *mailboxes* a serem processados. Para cada reclamação de *spam* em um *mailbox* especificado, são executados os seguintes passos:

1. Extrai a data de recebimento e obtém o respectivo dia;

[8] O limite máximo de agrupamento, ou mesmo a opção de não agrupar, é configurável.

2. Faz um busca nos registros de estatísticas gerais pelo dia obtido. Se não encontrar, cria um novo registro e inicializa seus contadores;

3. Incrementa o contador do total de reclamações. Identifica a categoria em que se enquadra a reclamação e incrementa o contador correspondente;

4. Se foi enviada pelo SpamCop e se é de algum tipo efetivamente tratado:

   (a) Extrai o endereço IP envolvido e verifica se pertence a alguma rede brasileira;

   (b) Faz um busca pelo registro de estatísticas por IP, por uma chave com o mesmo endereço IP e tipo. Se não encontrar, obtém os contatos responsáveis pelo IP e o bloco CIDR ao qual este IP pertence, cria um novo registro e o preenche com os dados obtidos;

   (c) Acrescenta a data de recebimento à lista de datas associadas, mantendo-a em ordem crescente.

Ao final da execução deste módulo, estão atualizados os conjuntos de registros de estatísticas gerais e de estatísticas por IP.

4.2.6 spamctl-stat-mgr

Este módulo manipula os arquivos de estatísticas gerais e de estatísticas por IP, e recebe como parâmetro o tipo de estatística a ser gerada.

Os relatórios gerados são apresentados na forma de tabelas, onde cada linha contém o número de reclamações de *spam* diárias para as diversas categorias e seu respectivo total. Um relatório também pode ser gerado apenas para um período específico.

Para os relatórios de estatísticas por IP, pode-se optar pela apresentação de dados agrupados por endereço IP, bloco CIDR ou responsável. Também é possível apresentar os dados de forma a gerar tabelas por responsável ou bloco CIDR das redes com maior número de reclamações associadas, apresentadas em ordem descresente do número de reclamações.

4.3 Aplicação de Métodos de Programação Segura

Todos os módulos foram implementados com especial atenção para diversos aspectos de programação segura [15], devido ao fato de manipularem dados potencialmente não confiáveis, enviados por terceiros via *email*.

Além da observação de aspectos de programação segura, foi adotado um recurso de segurança da linguagem Perl, denominado "*taint mode*" [16]. Neste modo, o interpretador Perl toma algumas precauções de segurança, tais como:

- marcar dados que provêm de fora do programa, como argumentos de linha de comando, variáveis de ambiente e conteúdo de arquivos, inicialmente como "*tainted*". Deste modo estes dados não podem ser usados em comandos que invoquem *shells*, modifiquem arquivos, diretórios ou processos. Para que eles possam ser utilizados, o programador precisa explicitamente selecionar um subconjunto válido da entrada de dados;

- obrigar que o programa defina explicitamente uma variável de ambiente PATH, e que os diretórios que o compõem tenham permissão de escrita apenas para o seu dono (*owner*) e grupo.

Também foram utilizados outros recursos de segurança do Perl, como a possibilidade de executar um programa externo sem a intervenção da *shell*, diminuindo assim os riscos com relação à interpretação de metacaracteres.

Além destes cuidados, os módulos também foram submetidos à análise da ferramenta de auditoria de código fonte RATS[9], à procura de problemas conhecidos de segurança.

5 RESULTADOS

Os resultados aqui apresentados, produzidos pelo sistema de controle e acompanhamento de notificações de *spam*, referem-se tanto às notificações que já foram enviadas para os responsáveis envolvidos, quanto aos relatórios contendo dados estatísticos extraídos das reclamações de *spam*.

5.1 Notificações Enviadas

O módulo responsável pelo envio das notificações entrou em produção em 30/06/2003.

Até o dia 30/07/2003 foram geradas 14.607 notificações, agrupadas em 957 mensagens, que foram, então, enviadas para os devidos responsáveis. Foram identificados 11.287 endereços IP distintos em tais notificações, provenientes de diversas redes brasileiras.

Alguns dos responsáveis que receberam as notificações enviadas têm se mostrado receptivos, interagindo com o grupo de resposta a incidentes que desenvolveu este sistema para solucionar seus respectivos problemas.

5.2 Relatórios com as Estatísticas Geradas

Para gerar os dados estatíticos extraídos de reclamações de *spam*, foi considerado um período diferente do apresentado na seção 5.1.

[9] `http://www.securesoftware.com/rats/`

Foram utilizados *mailboxes* contendo todas as reclamações de *spam* (1.550.476) recebidas entre os dias 01/01/2003 e 30/06/2003, correspondendo ao primeiro semestre de 2003.

Seguem os resultados gerados com base nos dados armazenados nos arquivos de estatísticas gerais e de estatísticas por endereço IP, para o período citado acima.

5.2.1 Estatísticas Gerais

O gráfico da Figura 3 foi criado com base em dados de estatísticas gerais, armazenados pelo sistema. Segue uma breve discussão sobre os dados observados neste gráfico.

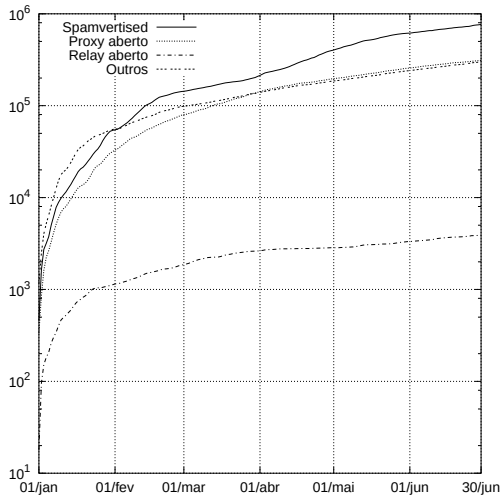

Figura 3: Totais reportados no primeiro semestre de 2003. Utilizada escala logarítmica.

Das reclamações envolvendo máquinas possivelmente sendo abusadas por terceiros, aquelas referentes a *proxies* abertos totalizaram 311.793, o que corresponde a aproximadamente 25%. Já as reclamações referentes a *relays* abertos somaram apenas 3.897, representando menos de 0,1% do total. Acredita-se que esta grande diferença, entre o número de reclamações do primeiro tipo em relação ao segundo, está relacionada com a popularização dos serviços de acesso à Internet via banda larga, como ADSL, *cable modem*, etc. As máquinas conectadas através de banda larga muitas vezes possuem um servidor *proxy* instalado com sua configuração padrão, que permite que o serviço seja abusado por terceiros [12].

Também é interessante observar que as reclamações do tipo *spamvertised website* somaram 767.282, correpondendo a 55% do total. Este não é um problema apenas técnico, mas sim uma questão de política de uso dos recursos da rede. Para resolvê-lo é necessário que os responsáveis por estas redes apliquem "Políticas de Uso Aceitável" que tenham regras claras sobre a utilização dos recursos e ações a serem tomadas em casos de abuso.

Confirmou-se, também, que a maior parte das reclamações recebidas, cerca de 80%, são dos tipos efetivamente tratados pelo sistema. Porém, existe um número expressivo (300.585) de reclamações não tratadas, que necessitam ser processadas no futuro.

5.2.2 Estatísticas por Endereço IP

O gráfico da Figura 4 foi criado com base em dados de estatísticas por IP, armazenados pelo sistema. Nele são apresentados os 10 blocos CIDR com o maior número de reclamações associadas, em ordem decrescente. Foi considerado o somatório das reclamações dos tipos *proxy* aberto, *relay* aberto e *spamvertised website*.

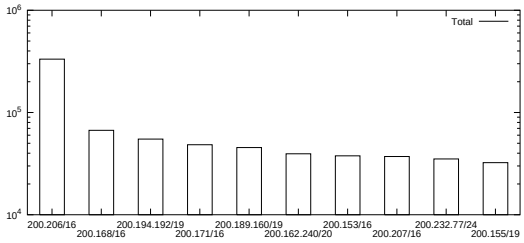

Figura 4: Redes com maior número de reclamações associadas, agrupadas por bloco CIDR. Utilizada escala logarítmica.

Dentre todas as reclamações associadas a esses 10 blocos CIDR, foram contabilizados cerca de 6000 endereços IP distintos. Para a grande maioria destes IPs foi possível determinar o nome completo de domínio (*Fully Qualified Domain Name*) a eles associados, através de uma consulta de DNS reverso. Mais da metade desses nomes sugerem que estes endereços IP estavam sendo utilizados na prestação de serviços envolvendo o acesso à Internet via banda larga (ADSL, *cable modem*, etc).

O fornecimento de relatórios agrupados por blocos CIDR pode auxiliar organizações a:

- relacionar reclamações de *spam* associadas a um determinado bloco CIDR sob sua responsabilidade com, por exemplo, um determinado serviço sendo prestado;
- focar seus esforços em partes de suas redes que apresentem o maior número de reclamações e, conseqüentemente, o maior número de possíveis problemas de segurança.

O sistema também permite gerar relatórios contendo dados agrupados por responsável de contato. A Figura 5 apresenta 3 gráficos, separados por tipo de reclamação efetivamente tratada pelo sistema, cada um contendo as 10 redes com o maior número de reclamações associadas.

As informações contidas nestes gráficos podem ser utilizadas não só para os mesmos propósitos re-

lacionados ao gráfico da Figura 4, mas também para que os responsáveis possam identificar quais problemas são mais iminentes e, até mesmo, estabelecer uma ordem de prioridade para combatê-los.

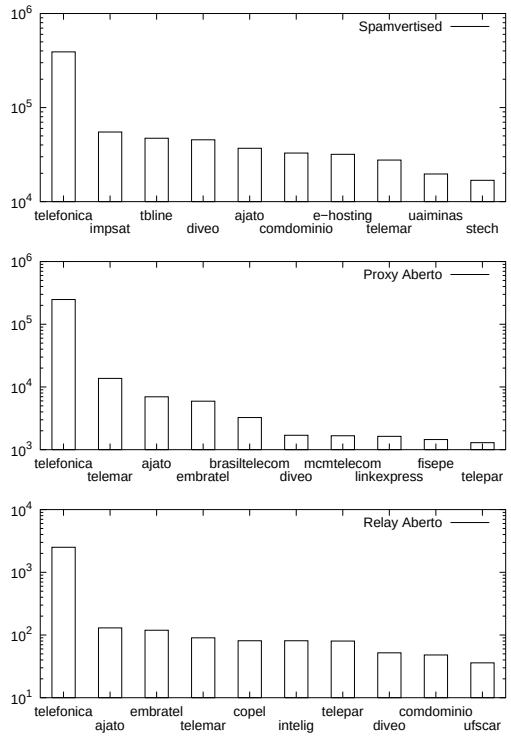

Figura 5: Redes com maior número de reclamações associadas, agrupadas por contato e separadas por tipo. Utilizada escala logarítmica.

6 TRABALHOS FUTUROS

O sistema de controle e acompanhamento de notificações de *spam* aqui apresentado trata apenas reclamações enviadas pelo SpamCop, de determinados tipos e que contém padrões bem definidos.

Desta forma, pretende-se expandir o sistema para que possa tratar reclamações enviadas pelo SpamCop, dos tipos que não foram efetivamente contemplados na atual implementação e, também, enviadas por outras fontes que, por ventura, forneçam serviço semelhante ao do SpamCop.

Além disso, planeja-se adequar e aplicar o sistema no controle e acompanhamento de reclamações envolvendo outras formas de atividades abusivas. Um exemplo seria aplicá-lo no tratamento de reclamações que discriminem eventos relacionados com *worms*.

7 CONCLUSÕES

A automatização de boa parte do processo de análise e tratamento de reclamações de *spam* tem permitido que o grupo de resposta a incidentes, onde o sistema foi implementado, realize um encaminhamento efetivo deste tipo de reclamação. O processamento manual de tais reclamações implicaria na alocação de diversas pessoas desse grupo para trabalhar em tempo integral na leitura dessas mensagens, e mesmo assim não asseguraria que fossem tratadas em tempo hábil. Deste modo, o sistema mostrou ser uma ferramenta eficiente e de grande importância para grupos de segurança, tratamento de abusos e resposta a incidentes que recebem este tipo de reclamação.

Devido ao curto intervalo de tempo desde o momento que o sistema entrou em produção, não foi possível avaliar se o envio de notificações para os responsáveis pelas redes envolvidas nas reclamações de *spam* colaborou para a redução destas atividades abusivas. Mas, espera-se que com o passar do tempo, as notificações enviadas por este ou por qualquer outro grupo que venha a utilizar o sistema, possam auxiliar na redução de tais atividades, bem como dos problemas de segurança relacionados.

As estatísticas geradas pelo sistema também podem ser utilizadas com os mesmos propósitos citados anteriormente. Num primeiro momento, a análise destas estatísticas pode direcionar os esforços, por parte dos responsáveis, para que busquem soluções para problemas sérios e emergenciais associados a suas redes. Num segundo momento, estas estatísticas podem ser utilizadas para avaliar se as reclamações, de determinado tipo de abuso e associadas a uma rede, reduziram ou até mesmo cessaram.

AGRADECIMENTOS

Os autores gostariam de agradecer a Frederico A. C. Neves e a Hugo K. Kobayashi, do Registro .br, e a Ricardo G. Patara, do LACNIC, pelo apoio dado a este projeto.

REFERÊNCIAS

[1] R. Shirey, “RFC 2828: Internet Security Glossary.” `http://www.ietf.org/rfc/rfc2828.txt`, May 2000.

[2] S. Hambridge and A. Lunde, “RFC 2635: DON’T SPEW – A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam).” `http://www.ietf.org/rfc/rfc2635.txt`, June 1999.

[3] R. Zakon, “RFC 2235: Hobbes’ Internet Timeline.” `http://www.ietf.org/rfc/rfc2235.txt`, November 1997.

[4] B. Hayes, “Spam, spam, spam, lovely spam,” *American Scientist*, vol. 91, pp. 200–204, May–June 2003.

[5] G. Lindberg, “RFC 2505: Anti-Spam Recommendations for SMTP MTAs.” `http://www.ietf.org/rfc/rfc2505.txt`, February 1999.

[6] T. Killalea, "RFC 3013: Recommended Internet Service Provider Security Services and Procedures." http://www.ietf.org/rfc/rfc3013.txt, November 2000.

[7] P. Graham, "A Plan for SPAM." http://www.paulgraham.com/spam.html, August 2002.

[8] D. Crocker, V. Schryver, and J. Levine, "Internet Draft: Technical Considerations for Spam Control Mechanisms." http://www.ietf.org/internet-drafts/draft-crocker-spam-techconsider-02.txt, May 2003.

[9] M. J. W. Brown, D. Stikvoort, K. P. Kossakowski, G. Killcrece, R. Ruefle, and M. Zajicek, *Handbook for Computer Security Incident Response Teams*. Carnegie Mellon University, 2nd ed., April 2003. CMU/SEI-2003-HB-002.

[10] M. Chatel, "RFC 1919: Classical versus Transparent IP Proxies." http://www.ietf.org/rfc/rfc1919.txt, March 1996.

[11] J. Klensin, "RFC 2821: Simple Mail Transfer Protocol." http://www.ietf.org/rfc/rfc2821.txt, April 2001.

[12] CERT Coordination Center, "Vulnerability Note VU#150227 – Multiple vendors' HTTP proxy default configurations allow arbitrary TCP connections via HTTP CONNECT method." http://www.kb.cert.org/vuls/id/150227, June 2003.

[13] B. W. Kernighan and R. Pike, *The Practice of Programming*. Addison-Wesley, 1999.

[14] L. Wall, T. Christiansen, and R. L. Schwartz, *Programming Perl*. O'Reilly & Associates, 2nd ed., 1996.

[15] J. Viega and G. McGraw, *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley, 2002.

[16] "Perl Programmers Reference Guide: perlsec – Perl security." http://www.perldoc.com/perl5.8.0/pod/perlsec.html.