SpSb: um ambiente seguro para o estudo de *spambots*

Gabriel C. Silva[1], Alison C. Arantes[2], Klaus Steding-Jessen[3], Cristine Hoepers[3], Marcelo H.P. Chaves[3], Wagner Meira Jr.[1], Dorgival Guedes[1]

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

[2]CSIRT/POP-MG – Equipe de resposta a incidentes de segurança do POP-MG

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

gabrielc@dcc.ufmg.br, alison@csirt.pop-mg.rnp.br
{jessen,cristine,mhp}@cert.br, {meira,dorgival}@dcc.ufmg.br

Resumo. Botnets *são consideradas a origem de grande parte do* spam *observado atualmente. Entretanto, a informação que se tem sobre esses sistemas costuma ser apócrifa ou deriva de esforços de engenharia reversa pontuais. Para se entender melhor o comportamento desses sistemas é necessário um ambiente de monitoração que dê ao* bot *a impressão de estar executando com liberdade na Internet, porém sem permitir que suas atividades causem dano à rede. Este artigo curto descreve uma implementação de tal sistema atualmente em curso.*

1. Introdução

No cenário atual de combate ao *spam* na Internet, *botnets* ocupam uma posição particularmente importante, por sua capacidade de disseminação de mensagens a partir de um grande número de máquinas comprometidas (*bots*) [Xie et al. 2008]. Um dos grandes desafios para a comunidade de segurança que combate *spam* é entender como essas redes operam, a fim de criar mecanismos de detecção e bloqueio dessas fontes de *spam* (denominadas *spambots*, nesse caso).

Na prática, a informação sobre o comportamento de *botnets* tende a ser apócrifa, sem validação científica, ou de validade limitada pela volatilidade da área. Dessa forma, é essencial que se tenha uma forma flexível de acompanhar o comportamento de novos *bots* à medida que eles surgem. Esse acompanhamento envolve tanto a coleta de binários de versões ativas de *malware* para análise, quanto o acompanhamento de sua execução.

O grande problema de se analisar o comportamento de um *bot* é que esse comportamento só pode ser observado em sua totalidade se lhe é garantido acesso à Internet. Como esses programas operam seguindo os comandos de um elemento central, denominado Comando-e-Controle (C&C), um *bot* só passa a atuar no envio de *spam*, por exemplo, após se conectar ao seu C&C e obter instruções sobre o conteúdo das mensagens e os destinatários. Entretanto, se o *malware* tem acesso à Internet, se torna difícil evitar que cause dano (por exemplo, enviando *spam*). Por esse motivo, serviços de análise de comportamento de binários, como o Anubis[1], se limitam a executar os binários sob suspeita sem permitir o seu acesso à rede, registrando apenas o seu comportamento na máquina local. Apesar de auxiliar na identificação do binário, essas informações não contribuem para o entendimento de seu comportamento na rede.

[1]http://anubis.iseclab.org/?action=sample_reports

O objetivo deste artigo é propor um ambiente seguro para o estudo de *spambots* que seja ser capaz de registrar o comportamento de todo o ciclo de vida de um *spambot* analisando seu tráfego de rede, sem permitir que ataques reais ocorram. Neste ambiente, qualquer *bot* sob análise deve ser capaz de trocar informações com seu C&C sem efetivamente conseguir enviar o *spam*. Pelas suas características, esse ambiente se encaixa na definição de um *sandbox*, daí o nome escolhido para o sistema, *SpSb* (*Spam Sandbox*).

2. Arquitetura proposta

Para garantir um ambiente seguro para análise de *malware*, pretendemos utilizar a infraestrutura de rede ilustrada na figura 1. A arquitetura inclui um sistema de captura de amostras de binários de possíveis *spambots* e o *sandbox* propriamente dito. Nesta seção discutimos os princípios de operação previstos para cada elemento do sistema. A seção seguinte discute detalhes de implementação relacionados.

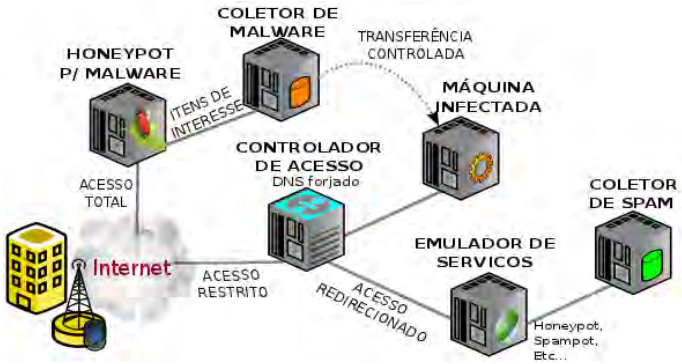

Figura 1. Arquitetura proposta

O sistema de captura inclui um *honeypot* especializado na coleta de *malware* e um serviço de avaliação, filtragem e armazenamento de amostras. O primeiro desafio do sistema é identificar as amostras de interesse. Para isso, cada amostra é comparada a outras já avaliadas e uma consulta é feita a serviços de análise externos. Apesar desses serviços não serem suficientes para determinar todo o comportamento de uma amostra, eles fornecem informações úteis, como a identificação de amostras claramente sem interesse nesse caso, como vírus e ou variações de outros *bots* já coletados. A transferência de uma amostra selecionada para o *sandbox* deve ser feita de forma bastante controlada, para evitar que a amostra contamine algum elemento de forma inesperada e para garantir que o mecanismo de transferência não possa vir a ser explorado por um *malware* durante sua execução no ambiente.

A máquina alvo que será infectada pode ser uma máquina real, ou uma máquina virtual executando em um dos ambientes de virtualização hoje disponíveis. A opção entre as duas opções representa um compromisso entre a flexibilidade de operação, a segurança do sistema e a gama de *malware* que poderão ser avaliados. O uso de máquinas virtuais simplifica a gerência e configuração do sistema, tornando simples recuperar uma visão de uma máquina em um determinado ponto de sua configuração, antes ou após a contaminação pelo *bot*. Entretanto, o gerente de máquinas virtuais está sujeito a ataques a partir da máquina virtual (ao menos potencialmente) e diversos tipos de software malicioso hoje incluem algum tipo de mecanismo para detectar sua instalação em uma máquina virtual [Miwa et al. 2007].

A tarefa de controlar a visão do *bot* para a Internet é dividida entre o controlador de acesso e o emulador de serviços. O primeiro é responsável por interceptar cada pacote gerado pela máquina infectada e decidir sua destinação, que deve se encaixar em uma das opções a seguir: (i) processar o pacote localmente, (ii) permitir que o pacote siga seu caminho para a Internet, (iii) redirecionar o pacote para um emulador de serviços, ou (iv) descartar o pacote. Pacotes como consultas DNS devem ser tratadas diretamente pelo controlador, A interceptação desse protocolo tem dois objetivos: observando o padrão de consultas DNS de um *bot* é possível, em muitos casos, inferir a natureza do seu C&C [Choi et al. 2009]; além disso, caso algum tráfego seja identificado com um ataque iniciado pelo *bot* (como o envio de *spam*), o servidor de DNS do controlador de acesso tem a informação necessária para redirecionar o tráfego para o emulador de serviços. Isso pode ser feito pela re-escrita dos endereços de resposta, ou pela observação do endereço de resposta e pela inserção de regras de roteamento que interceptem o tráfego para aqueles endereços e os redirecionem para o emulador.

Pacotes identificados como associados ao fluxo de controle do *bot*, direcionados principalmente ao seu Comando-e-Controle, devem ser encaminhados normalmente pela Internet. Essa identificação pode ser feita através dos padrões de DNS, como mencionado, pelo tipo de protocolo utilizado (p.ex., IRC) e pelo momento da conexão. Para evitar problemas devido a ataques que porventura possam ser encapsulados em consultas aparentemente inócuas, um controle rigoroso de taxa de comunicação deve ser implementado. Tráfego redirecionado para o emulador de serviços inclui todo tipo de protocolo que seja classificado com parte de um ataque. Entre esses protocolos destacam-se ataques a outras máquinas com o objetivo de propagar o *malware* e tentativas de distribuição de *spam*. Como mencionado anteriormente, o emulador de serviços deve manter uma comunicação constante com o controlador de acesso, pois ele deve ser informado sobre como responder a cada requisição redirecionada (por exemplo, uma conexão SMTP redirecionada deve ser respondida com o nome do servidor alvo pretendido, bem como seu endereço IP). Essa informação deve ser repassada pelo controlador de acesso. As mensagens de *spam* coletadas são armazenadas e processadas para extrair informações que permitam classificá-las.

Um objetivo importante do *Spam Sandbox* é automatizar as decisões sobre que tráfego do *bot* pode acessar a Internet e que tráfego deve ser direcionado para o emulador. Por exemplo, uma sequência desse tipo poderia ser vista da seguinte forma a partir do controlador de acesso: uma consulta DNS é seguida por uma conexão ao porto 6667 (IRC) do endereço IP fornecido pela resposta DNS — o controlador permite a conexão e a troca de tráfego com o destino, por se tratar de uma conexão ao C&C; uma nova consulta DNS é seguida por uma conexão ao porto 25 (SMTP) do novo endereço — nesse caso, o controlador notifica o emulador sobre o nome e o IP que o *bot* tenta conectar e passa a redirecionar o tráfego para aquele IP para o emulador, que passa a executar o código do emulador de um servidor de correio, armazenando a mensagem de *spam* que o *bot* acha que foi entregue.

3. Aspectos de implementação

O sistema está sendo implementado utilizando módulos disponíveis na comunidade de sofwtare livre e aberto, bem com módulos especialmente desenvolvidos para esse fim. O

sistema de coleta de amostras de *malware* utiliza o *honeypot* Dionaea[2]. As amostras coletadas são enviadas para análise pelo serviços Anubis[3] e Norman Sandbox[4]. No momento, a análise da informação obtida dessas fontes é manual. No futuro, o processamento será automatizado com extratores automáticos, para simplificar a coleta de amostras.

Nossa primeira opção para a máquina infectada é a utilização de uma máquina virtual executando no ambiente Virtual Box. O ambiente é configurado de forma a remover elementos de configuração da máquina virtual que são usualmente utilizados por *bots* para determinar se o ambiente é uma máquina virtual. Dessa forma, podemos nos beneficiar dos recursos de manipulação de imagens e armazenamento de *snapshots* para retornar o sistema a uma configuração pré-determinada. A inserção de *malware* é feita atualmente através de um dispositivo USB; estamos estudando o ambiente virtualizado para poder configurar a amostra diretamente na imagem da máquina virtual.

O controlador de acesso é implementado com uma máquina FreeBSD, utilizando o sistema de filtragem de pacotes nativo daquele sistema, o PF (*BSD Packet Filter*). As interfaces da máquina são conectadas através de uma *switch* virtual transparente configurada pelo sistema operacional. O processo de captura e re-escrita de pacotes é feito com regras PF, com um tratamento especial dos pacotes de retorno do emulador de serviços para garantir o roteamento correto de pacotes com endereços forjados. O software de controle do PF está sendo desenvolvido para interceptar os pacotes recebidos, tomar decisões sobre os próximos passos, inserir novas regras para determinar como as novas conexões serão tratadas e notificar o emulador a respeito dos serviços que ele deve emular.

O emulador de serviços contém um segundo servidor Dionaeae um *honeypot* especialmente desenvolvido para a coleta de *spam* [Steding-Jessen et al. 2008]. Ambos os servidores estão sendo configurados para se adequarem ao ambiente emulado. Em particular, eles devem receber comandos do controlador de acesso para saberem quais endereços IPs devem ser emulados e qual o nome o servidor deve ser usado em cada caso. O coletor de *spam* é basicamente o mesmo desenvolvido originalmente para aquele *honeypot*. Outros serviços podem ser incluídos posteriormente.

Técnicas de análise do *spam* coletado estão sendo desenvolvidas para identificar os padrões de tráfego das *botnets* e os padrões de máquinas alvo que seriam atacadas pelo *bot* caso ele operasse na Internet. Essas técnicas são desenvolvidas em conjunto com outros trabalhos de análise de *spam* atualmente em atividade no grupo do DCC/UFMG em conjunto com o CERT.br e o CSIRT/POP-MG [Guerra et al. 2010].

4. Trabalhos relacionados

Muitos trabalhos se orientam por princípios gerais universalmente reconhecidos a respeito de *botnets*, sem entretanto oferecer uma confirmação científica para esses princípios. Por exemplo, ao assumir que todas as conexões que chegam a um servidor de correio eletrônico tentando entregar mensagens de *spam* se originam de *botnets*, Xie et al. identificam tais redes com base nos endereços de origem das conexões contendo *spam*, sem uma confirmação direta de sua natureza [Xie et al. 2008]. Outro trabalho que se orienta por esses princípios gerais é o da ferramenta de detecção BotGad [Choi et al. 2009]. Nesse

[2] http://dionaea.carnivore.it/
[3] http://anubis.iseclab.org/
[4] http://www.norman.com/security_center/security_tools/

caso, um grupo de máquinas com certos comportamentos comuns na rede (p.ex., consultas DNS semelhantes) é identificado como uma *botnet*.

Os trabalhos mais próximos desta proposta são Botlab [John et al. 2009] e *Mimetic Internet* [Miwa et al. 2007]. Botlab propõe uma arquitetura de análise segura, mas depende diretamente da intervenção de um operador, que deve decidir como reagir a cada ação do *bot* sendo observado. Já *Mimetic Internet* propõe um arcabouço isolado que tenta reproduzir a Internet, com servidores que simulam sites populares, por exemplo, mas que não permite nenhum acesso à Internet real. Nesse caso, um *spambot* não seria capaz de contactar o restante da sua rede, o que limitaria sua ação.

5. Conclusão e trabalhos futuros

Observar o comportamento de uma *botnet* em uma campanha de *spam* a partir de um de seus componentes pode gerar um grande volume de informações sobre esses sistemas e formas de evitar sua ação. Entretanto, realizar esse estudo exige que se permita que um *spambot* acesse a Internet para estabelecer contato com seu comando-e-controle e se convencer de que está executando livremente, ao mesmo tempo que se evite que o mesmo cause danos reais à rede (p.ex., pelo envio de *spam*).

A arquitetura descrita neste artigo visa oferecer condições para que essa análise seja possível, de forma bastante automatizada. A combinação de um filtro de pacotes altamente flexível, com capacidade de redirecionamento e re-escrita de pacotes, e um conjunto de emuladores de serviços operando de forma integrada a esse filtro pode permitir que pacotes/fluxos identificados como de controle possam ter permissão para acessar a Internet, enquanto comportamentos daninhos, como o envio de *spam*, podem ser direcionados para servidores apropriados.

A implementação desse sistema se encontra em curso. Apesar de já concebermos uma solução completa, há ainda muito espaço para pesquisa no desenvolvimento de métodos para identificação de padrões de controle e de ataques e para o aumento do grau de automatização dos mecanismos de redirecionamento e emulação de serviços.

Referências

Choi, H., Lee, H., and Kim, H. (2009). Botgad: detecting botnets by capturing group activities in network traffic. In *Proceedings of the Fourth International ICST COMSWARE*, pages 1–8, New York, EUA. ACM.

Guerra, P. H. C. et al. (2010). Exploring the spam arms race to characterize spam evolution. In *Proceedings of the 7th CEAS*, Redmond, WA.

John, J. P. et al. (2009). Studying spamming botnets using botlab. In *Proceedings of the 6th USENIX NSDI*, pages 291–306.

Miwa, S. et al. (2007). Design and implementation of an isolated sandbox with mimetic internet used to analyze malwares. In *Proceedings of the USENIX DETER Workshop on Cyber Security Experimentation and Test*, pages 1–9.

Steding-Jessen, K. et al. (2008). Using low-interaction honeypots to study the abuse of open proxies to send spam. *Infocomp (UFLA)*, 7:44–52.

Xie, Y. et al. (2008). Spamming botnets: signatures and characteristics. *SIGCOMM CCR*, 38(4):171–182.