Arquitetura de um sistema integrado de defesa cibernética para detecção de *botnets*

Sérgio dos Santos Cardoso Silva[1] **e Ronaldo Moreira Salles**[1] **(IME)**
Praça General Tiburcio, 80 – 22290-270 – Rio de Janeiro – RJ – Brasil

{**scardoso,salles**}**@ime.eb.br**

[1]Programa de Engenharia de Defesa – Instituto Militar de Engenharia

Abstract. *This work proposes a novel architecture for for the design of a cyber defense system able to detect bots and block communication between bots and botnets. It is also proposed an algorithm based on graphs for detecting bots. The architecture was tested using DNS logs of an academic institute. Experiments were carried out by analyzing records of DNS queries in order to find machines suspected of being zombies. Preliminary results show that mechanisms employed can filter DNS records and identify machines suspected of belonging to a botnet.*

Resumo. *Este trabalho tem por objetivo apresentar uma proposta de arquitetura para um sistema de defesa cibernética capaz de detecar* bots *e bloquear a comunicação entre os bots e as* botnets*. Também é proposto um algoritmo de detecção de* bots *baseado em grafos de relacionamento. Foram realizados experimentos por meio da análise de registros de consultas DNS com o objetivo de encontrar máquinas suspeitas de serem zumbis. Resultados preliminares mostram que os mecanismos empregados permitem filtrar os registros de DNS e identificar máquinas suspeitas de pertencerem a uma botnet.*

1. Introdução

Diversos tipos de ataques cibernéticos contra estruturas críticas e usuários comuns têm sido relatados nos últimos anos, trazendo à tona a preocupação do emprego desses ataques em ações de Guerra Cibernética. Em algumas dessas ações foram empregadas estruturas conhecidas como *botnets*. As *botnets* são redes formadas por *hosts* escravos, denominados *bots*, que são controladas por um ou mais atacantes, denominados *botmasters*, que tem por objetivo realizar algum tipo de ação maliciosa [Rajab et al. 2006, Zeidanloo et al. 2010].

Atualmente, as *botnets* são uma das principais ameaças ao funcionamento dos serviços de rede [Ceron et al. 2010], pois permitem ao atacante que controla essas máquinas realizar diversas ações maliciosas, como roubo de informações, envio de mensagens não solicitadas (*spam*), ataques de negação de serviços, manipulação de jogos ou pesquisas online, entre outras.

Sendo assim, é necessário que sejam realizados esforços para combater essas redes, identificando as máquinas componentes e desativando a comunicação entre elas. Entretanto, realizar a detecção da *botnet* é uma atividade difícil. As *botnets* utilizam técnicas que permitam ficar a maior parte do tempo ativas, ocultando o tráfego gerado pelas botnets, tornando-as invisíveis [Dagon et al. 2007]. Um exemplo desses mecanismos é a redução do número de trocas de mensagens entre os componentes da rede.

Os trabalhos existentes sobre *botnets* normalmente descrevem soluções isoladas apresentando mecanismos de detecção [Gu et al. 2008, Coskun et al. 2010, Zhang et al. 2011] ou técnicas para bloqueio e desativação das *botnets* [Stone-Gross et al. 2011, Sanchez et al. 2011]. Além disso, as soluções comerciais existentes [Snort 2006] empregam técnicas baseadas em assinaturas, que tem limitação em detectar novos *malwares* ou *bots*, devido as caracteríticas furtivas que estes possuem.

Considerando a pesquisa bibliográfica realizada, não existe uma proposta de solução integrada que permita tanto a detecção quanto a desativação das *botnets*. Portanto, este trabalho tem por objetivo apresentar uma proposta de arquitetura de sistema de defesa cibernética capaz de realizar a detecção de máquinas *bots* e bloquear a comunicação entre os bots e as *botnets*, diminuindo a sua efetividade e, possivelmente, desarticulando-as. Também é proposta uma técnica de detecção de *bots* que utilizam o protocolo DNS. Resultados preliminares mostram que os mecanismos empregados permitem filtrar os registros de DNS e identificar máquinas suspeitas.

As principais contribuições desse artigo são:

- Apresentar uma proposta de arquitetura para um sistema integrado de defesa cibernética.
- Propor uma técnica de detecção de *bots* baseada em análise de logs DNS, empregando a arquitetura proposta.

Este trabalho está organizado conforme descrito a seguir. Na Seção 2 é apresentado o ciclo de vida de um *bot*. Na Seção 3 é apresentada a arquitetura de um sistema integrado de defesa cibernética e o estudo de caso para o DNS na seção 4. A Seção 5 apresenta os resultados iniciais com a análise de registros de acesso ao DNS. Por fim, a Seção 6 mostra as conclusões e possibilidades de trabalhos futuros.

2. Ciclo de vida dos *bots*

Para um melhor entendimento do problema e das técnicas propostas, será apresentado o ciclo de vida de um *bot*, que são as etapas que um *host* deve realizar para que se se torne um *bot* e pertença a uma *botnet* [Zhu et al. 2008]. Estas etapas podem receber nomes diferentes, mas os mais adotados estão ilustrados na figura 1.

A infecção inicial é o primeiro passo para transformar uma máquina vulnerável em um *bot*. O invasor realiza uma varredura em uma sub-rede buscando vulnerabilidades conhecidas e infecta a vítima por diversos métodos, por exemplo, através de *exploits*, anexos de mensagens, entre outros [Rajab et al. 2006, Dagon et al. 2007].

Após o invasor infectar a máquina, esta passa para a fase de Injeção Secundária. Nesta etapa, a máquina infectada executa um *script* que busca o código do *malware* em um repositório, transformando efetivamente essa máquina em um *bot* [Rajab et al. 2006].

Após a máquina se tornar um *bot*, ela contacta o servidor de comando e controle (C&C), processo denominado *rally* ou conexão. Esta etapa irá ocorrer toda vez que a máquina infectada for reiniciada, pois, é necessário que ela contate o C&C para informar a sua participação na *botnet*, assim como, para receber os comandos para realização de alguma atividade maliciosa [Dagon et al. 2007].

Devido a essa necessidade de contato com os servidores de C&C, essa fase pode ser considerada como uma etapa vulnerável da *botnet*, principalmente porque existirá um

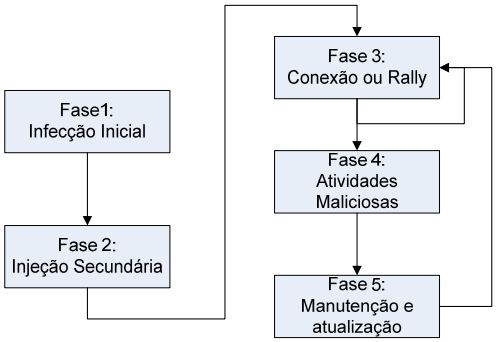

Figura 1. Ciclo de Vida dos Bots

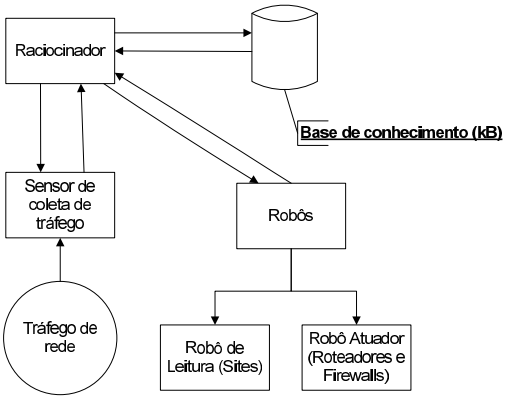

Figura 2. Arquitetura de sistema integrado de defesa cibernética

comportamento padrão dos *bots* para realizar a conexão, permitindo que sejam criados mecanismos que possam identificar esse padrão de tráfego e, consequentemente, identificar os componentes da *botnet*.

Após estabelecido o canal de comando e controle, o *bot* aguarda por comandos para realizar alguma atividade maliciosa [Rajab et al. 2006]. A partir dessa fase o *bot* está pronto para realizar algum tipo de ataque.

A última fase do ciclo de vida dos *bots* é a manutenção e atualização dos *malwares* utilizados. A manutenção está associada ao fato de que o *botmaster* deseja manter o seu exército de *bots*. Para isso, é necessário atualizar os códigos por diversos motivos, tais como, evadir das técnicas de detecção, adicionar novas funcionalidades ou migrar para outro servidor de C&C [Zhu et al. 2008].

3. Arquitetura do Sistema

Devido à gravidade do problema é necessário o desenvolvimento de soluções que possam integrar as duas fases importantes para combater as *botnets*: detecção e mitigação. A arquitetura de sistema integrado de defesa cibernética proposta na figura 2 integra essas duas fases necessárias ao tratamento do problema detectando a atividade maliciosa e realizando alguma medida para mitigar o problema.

Esta arquitetura é desenvolvida de forma modular permitindo, além da análise do tráfego da rede, a análise semântica dos dados das máquinas atacantes e atacadas, facilitando a identificação das máquinas *bots* e a adoção de medidas de proteção necessárias. A modularidade também permite uma visão abrangente dos vários componentes das redes de computadores monitoradas, fornecendo diferentes funcionalidades, dependendo da perspectiva considerada. A arquitetura é formada pelos seguintes componentes:

- **Sensor de coleta de tráfego**: monitora de forma passiva o tráfego de rede, fornecendo informações importantes na detecção e prevenção de ataques. Este componente está relacionado com as funções de sensoriamento do sistema;
- **Raciocinador**: executa funções voltadas para a análise de informações, processando e analisando os dados coletados, considerando aspectos estatísticos e semânticos dessas informações. A inteligência da arquitetura considera a análise

do tráfego aliada à análise semântica como uma forma de aprimorar a detecção e proteção contra ataques;

- **Base de conhecimento (kb)**: componente responsável por armazenar aspectos estatísticos e semânticos do tráfego de rede que são utilizados pelo raciocinador na análise das informações;
- **Robôs de atuação e leitura**: são responsáveis pelas funções de proteção. Podem alterar o funcionamento de dispositivos da rede de modo a melhorar a sua proteção, evitando potenciais ataques, ou coletar informações adicionais para confirmar algum tipo de atividade maliciosa.

O tráfego de rede analisado pelas funções de sensoriamento pode ser obtido de forma bruta, através da coleta de registros de tráfego (*traces*), com algum nível de agregação (fluxos) ou análise de registros (*log*). Para cada tipo de registro a ser analisado é implementado um módulo específico de sensoriamento e coleta de dados.

Na função de análise de informações é empregado um raciocinador modular que emprega algoritmos específicos para análise dos dados obtidos pelo sensor de coleta de tráfego. A partir do resultado dos algoritmos empregados, o raciocinador irá acionar robôs de leitura ou atuação para realizar alguma atividade complementar. O disparo dos robôs é baseado em valores pré-definidos armazenados na base de conhecimento do sistema.

Os robôs são agentes de softwares que realizam atividades complementares ao raciocinador. O robô de leitura pode ser acionado com o objetivo de obter informações adicionais necessárias a tomada de decisão do raciocionador. Por exemplo, se um determinado domínio foi considerado suspeito pelo algoritmo de análise de DNS, o raciocinador pode ativar o robô de leitura para obter mais informações, como, *PageRank* ou dados semânticos do *site*, para poder confirmar que aquele domínio é um C&C. Essas informações adicionais são adicionadas à base de conhecimento.

Confirmado que determinado domínio e/ou máquina é um *bot* ou C&C, o raciocinador pode, então, acionar o robô de atuação que irá adotar alguma medida para mitigar o problema. Os robôs de atuação podem ser especializados, por exemplo, como robôs para *firewalls*, servidores e roteadores. Um exemplo de medida preventiva que pode ser tomada é a adição de uma regra em um *firewall* com o objetivo de bloquear, exclusivamente, a comunicação entre o *bot* e o centro de comando e controle.

4. Estudo de caso - DNS

O primeiro módulo implementado na arquitetura proposta é para análise de registros de acesso (*log*) DNS. As consultas DNS são necessárias e importantes na *botnet*, especialmente nas centralizadas, porque durante a fase de *rally* o *bot* precisa encontrar o C&C e, para isso, precisa traduzir o domínio especificado para o centro de comando e controle.

Os registros de acesso obtidos pelo sensor de coleta são mapeados em Grafos de Contatos, modelados através de um grafo direcionado bipartido e ponderado $Gdns = (U, V, E, P)$, cujos nós são formados pelos endereços IP (U) e os domínios (V). Existirá uma aresta entre u ($u \in U$) e v ($v \in V$) se u solicitar uma consulta DNS para o domínio v. A quantidade de consultas determinará o peso das arestas (P).

O objetivo é a partir do grafo de contato identificar sub-grafos que formam comunidades de interesses suspeitas, permitindo encontrar o conjunto de *bots* que pertençam

a esta *botnet*. A identificação das comunidades deve ser baseada nas características dos *bots*, mapeadas nas propriedades dos grafos através da adoção das seguintes premissas:

I. **Interesse comum**: Os *bots* normalmente infectam mais de uma máquina na mesma rede [Coskun et al. 2010]. Essa caracterítica é mapeada através da análise do grau de entrada do nó de destino, que deve ser maior que 2 ($\theta \geq 2$);
II. **Comportamento robótico**: O *bot* é um *software* pré-programado, então a comunicação dos *hosts* infectados com a *botnet* terá um comportamento semelhante, por exemplo, realizando o mesmo número de consultas. Essa característica é mapeada através da análise dos pesos das arestas que incidem em determinado domínio. O comportamento robótico será caracterizado pelo desvio padrão (σ) do peso das arestas (P) estar abaixo de um determinado limiar ($\sigma \leq \alpha$) ou ser igual a zero;
III. ***Stealthiness***: Com o objetivo de evadir dos mecanismos de detecção, os *bots* empregam mecanismos que tornam sua comunicação de difícil detecção, realizando poucas consultas a muitos domínios distintos. Essa característica pode ser mapeada através do peso baixo das arestas que incidem em determinado domínio ($P \leq \beta$).

Com base nessas premissas foi implementado no raciocinador um algoritmo para análise do grafo $Gdns$ composto pelos seguintes passos:

1. Filtragem inicial;
2. Separação em subgrafos desconexos para cada domínio v ($v \in V$);
3. Busca por domínios que possuem arestas incidentes com desvio padrão baixo ($\sigma \leq \alpha$);
4. Busca por arestas com baixo peso ($P \leq \beta$).

A filtragem inicial tem por objetivo reduzir o tamanho do grafo e otimizar a busca por domínios suspeitos, removendo aqueles que com certeza não são C&C e que não se enquadram na premissa (I). A remoção foi realizada através da aplicação das seguintes regras da base de conhecimento:

- Aplicar uma *whitelist* com o objetivo de retirar os nós que não são C&C, por exemplo, domínios $gTLD$ .gov, .edu, .mil; $ccTLD$ .gov.br, .edu.br, .mil.br, .eb.br; ou domínios legítimos, como, facebook.com, google.com, entre outros;
- Retirar os nós de destino que possuem grau de entrada menor que 1 ($\theta \leq 1$);

O grafo de contatos filtrado ($Gdns' = (U, V, E, P)$) é separado em subgrafos desconexos para cada domínio v ($v \in V$). O subgrafo representa a comunidade de nós que acessam determinado domínio. Com o objetivo de mapear a característica robótica de um *bot*, são mantidos apenas os subgrafos que possuem arestas com baixa variação no número de contatos ($\sigma \leq \alpha$), de acordo com a premissa (II). O valor de α é obtido da base de conhecimento.

Os subgrafos resultantes da etapa (3) indicam nós suspeitos de pertencerem a uma *botnet*, porém, ainda podem existir nós que não realizem atividade maliciosa, mas se enquadram nessas características. Um exemplo, são as consultas realizadas para domínios que são contadores de acesso a páginas *web* (http://statcounter.com/) que possuem comportamento semelhante, pois, também são *softwares* pré-programados.

Com o objetivo de remover os domínios que não são *bots* é aplicada a etapa (4) do algoritmo, baseada na premissa (III). São removidos todos os subgrafos que possuem arestas com pesos superior a β. O valor de β é obtido da base de conhecimento.

Os subgrafos resultantes das 4 etapas do algoritmos contém os endereços IP e domínios suspeitos de pertencerem a uma *botnet*. Para uma confirmação, o raciocinador aciona o robô de leitura com o objetivo de confirmar se determinado subgrafo é uma comunidade *botnet*. O robô pode confirmar, por exemplo, através da análise semântica do site ou da verificação se determinado domínio está em uma *blacklist*.

Confirmada a participação, o raciocinador aciona o robô de atuação para realizar alguma ação de bloqueio, por exemplo, criando uma regra no *firewall*, removendo o domínio, comunicando o administrador da rede, entre outras.

5. Resultados Preliminares

Os primeiros resultados foram obtidos com a análise do *Log* DNS de uma instituição de ensino, no período de período de 07/07 à 19/07/2011 e 24/02 à 04/04/2012. Entretanto, devido a limitação de espaço para o artigo serão apresentados resultados para 3 dias.

A tabela 1 apresenta um resumo do número de endereços IP e domínios distintos presentes no *log*, de acordo com cada fase do algoritmo implementado no raciocinador. Como pode ser verificado, após a etapa (1) houve uma redução significativa, superior a 97%, do número de endereços IP e domínios distintos.

Para determinar os valores para α e β aplicados nas etapas (3) e (4), respectivamente, realizou-se uma busca de domínios suspeitos nos acessos ao servidor de DNS. Foram considerados domínios suspeitos os que pertenciam ao $ccTLD$ .cn, de acordo com [Rajab et al. 2006], 95% destes domínios estão infectados, e ao .ws, devido a não ser comum um acesso a esse site considerando acessos iniciados no Brasil.

A partir dos domínios considerados suspeitos foi adotada a mediana e o desvio padrão do número de contatos para determinar os valores de corte $\alpha = 0.3$ e $\beta = 1.3$. O número de endereços e domínios restantes após cada fase são apresentados na tabela 1.

Tabela 1. Número de endereços e domínios distintos para cada fase do algoritmo

Etapa	Dia 12/03/12		Dia 13/03/12		Dia 14/03/12	
	# IP	# Dom	# IP	# Dom	# IP	# Dom
Gdns	20.529	238.852	30.027	383.977	37.355	499.777
1	540	6.962	784	11.428	949	15.040
2	540	6.962	784	11.428	949	15.040
3	290	1.359	260	2.244	122	2.918
4	245	1.149	221	1.899	104	2.492

Para confirmar se os hosts e os domínios pertenciam ou não a uma *botnet* foi realizada uma verificação dos sites e foi detectado que 5 endereços IPs e 253 domínios pertenciam à *botnet* Conficker. Dos 5 endereços encontrados, 3 pertencem a mesma subrede (200.213.8y.xxx) e os outros dois endereços são de subredes distintas (200.20.12y.xxx e 200.20.17z.kkk). Os domínios consultados não possuem um nome que apresente um significado relevante, por exemplo, *vbhtzo.ws* e *vqtqq.cn*.

A vantagem da solução está na taxa de acerto para os domínios e endereços suspeitos, pois, foram todos detectados. Entretanto, a desvantagem da solução está relacionada a alta taxa de falsos positivos, superior a 90%. O aprimoramento na criação da *whitelist* contribuirá para a redução desse valor.

6. Conclusão

Os resutlados preliminares demonstraram que a arquitetura e o algoritmo proposto para o protocolo DNS apresentaram resultados satisfatórios na detecção de máquinas e domínios suspeitos. Entretanto são necessários estudos para aprimorar a obtenção da *whitelist*, com o objetivo de reduzir significativamente a taxa de falso positivos; para a determinação de α e β com o objetivo de aumentar a precisão na detecção dos *bots*; e para a criação de mecanismos para permitir ao robô de leitura confirmar se um determinado domínio ou endereço suspeitos pertencem a uma *botnet*.

Referências

Ceron, J. M., Granville, L. Z., and Tarouco, L. M. R. (2010). Uma arquitetura baseada em assinaturas para mitigação de botnets. In *X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 105–118.

Coskun, B., Dietrich, S., and Memon, N. (2010). Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. In *Proc. of the 26th Annual Computer Security Applications Conference*, ACSAC 10, New York. ACM.

Dagon, D., Gu, G., Lee, C. P., and Lee, W. (2007). A taxonomy of botnet structures. In *Computer Security Applications Conf., 2007. ACSAC 2007. 23th Annual*.

Gu, G., Perdisci, R., Zhang, J., and Lee, W. (2008). BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proc. of the 17th Conf. on Security symposium*, pages 139–154, Berkeley, CA, USA. USENIX.

Rajab, M. A., Zarfoss, J., Monrose, F., and Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, page 41–52, New York. ACM.

Sanchez, F., Duan, Z., and Dong, Y. (2011). Blocking spam by separating end-user machines from legitimate mail server machines. In *Proc. of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS '11, New York. ACM.

Snort (2006). Snort. http://www.snort.org.

Stone-Gross, B., Cova, M., Gilbert, B., Kemmerer, R., Kruegel, C., and Vigna, G. (2011). Analysis of a botnet takeover. *Security Privacy, IEEE*, 9(1):64 –72.

Zeidanloo, H. R., Shooshtari, M. J., Amoli, P. V., Safari, M., and Zamani, M. (2010). A taxonomy of botnet detection techniques. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE Inter. Conf. on*, volume 2, pages 158 –162.

Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., and Luo, X. (2011). Detecting stealthy P2P botnets using statistical traffic fingerprints. DNS 2011, pages 121–132, Los Alamitos, CA, USA. IEEE Computer Society.

Zhu, Z., Lu, G., Chen, Y., Fu, Z. J., Roberts, P., and Han, K. (2008). Botnet research survey. In *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, pages 967 –972.

