

Detecção de *Botnets* através da Análise do tráfego DNS e Engenharia Reversa

Juliano Stolpe

Universidade Regional Integrada do Alto Uruguai e das Missões (URI)
Departamento de Engenharias e Ciência da Computação – Santo Ângelo, RS - Brasil.

jstolpe.nti@gmail.com

Abstract. Some botnets uses dynamic addressing techniques such as ip-flux and domain-flux for communication between bots and the command and control server, thus making them more robust for operation and therefore more difficult for detection. This paper details a method to perform botnets detection through DNS traffic analysis along with reverse engineering of malicious code. Such method is a set of procedures to be followed for identification of hosts characterized as bots. Two case studies were presented that were successful with the application of this method.

Resumo. Algumas botnets utilizam-se de técnicas de endereçamento dinâmico como ip-flux e domain-flux para a comunicação entre os bots e o servidor de comando e controle, tornando-as assim mais robustas para a operação, e consequentemente mais difíceis para a detecção. Este artigo detalha um método para realizar a detecção de botnets através da análise do tráfego DNS junto com a engenharia reversa de código malicioso. O método é um conjunto de procedimentos que devem ser seguidos para a obtenção de hosts caracterizados como bots. São apresentados dois estudos de caso que obtiveram êxito com a aplicação deste método.

1. Introdução

Devido ao acesso livre e distribuído do protocolo DNS, aplicações maliciosas também podem fazer consultas para realizar ataques, dentre elas *botnets* que podem ser definidas como um conjunto de máquinas comprometidas que permitem ao atacante o controle remoto dos recursos computacionais para realizar atividades fraudulentas ou ilícitas [McCarty 2003b, Freiling et al. 2005]. Tais máquinas utilizam um software chamado de *bot* (da palavra robô), o qual liga os computadores infectados a uma infraestrutura de Comando e Controle (C&C).

Alguns trabalhos propuseram sistemas, ferramentas e arquiteturas para detecção e mitigação de *botnets*, como [Ceron J. 2010] que definiu uma arquitetura baseada em assinatura de rede de máquinas comprometidas por bots, [Laufer 2005] propôs um sistema de rastreamento de pacotes para descobrir a origem de ataques, [Hossain 2010] propôs a mineração do tráfego DNS para detecção de aplicações de envio de Spam, e [Kaio 2014] apresenta uma metodologia utilizando teoria dos grafos para distinguir consultas padrões de anômalas no tráfego DNS.

A Tabela 1 faz uma compilação dos trabalhos relacionados ao tema, demonstrando as diferenças e semelhanças entre os trabalhos relacionados para o desenvolvimento do método proposto.

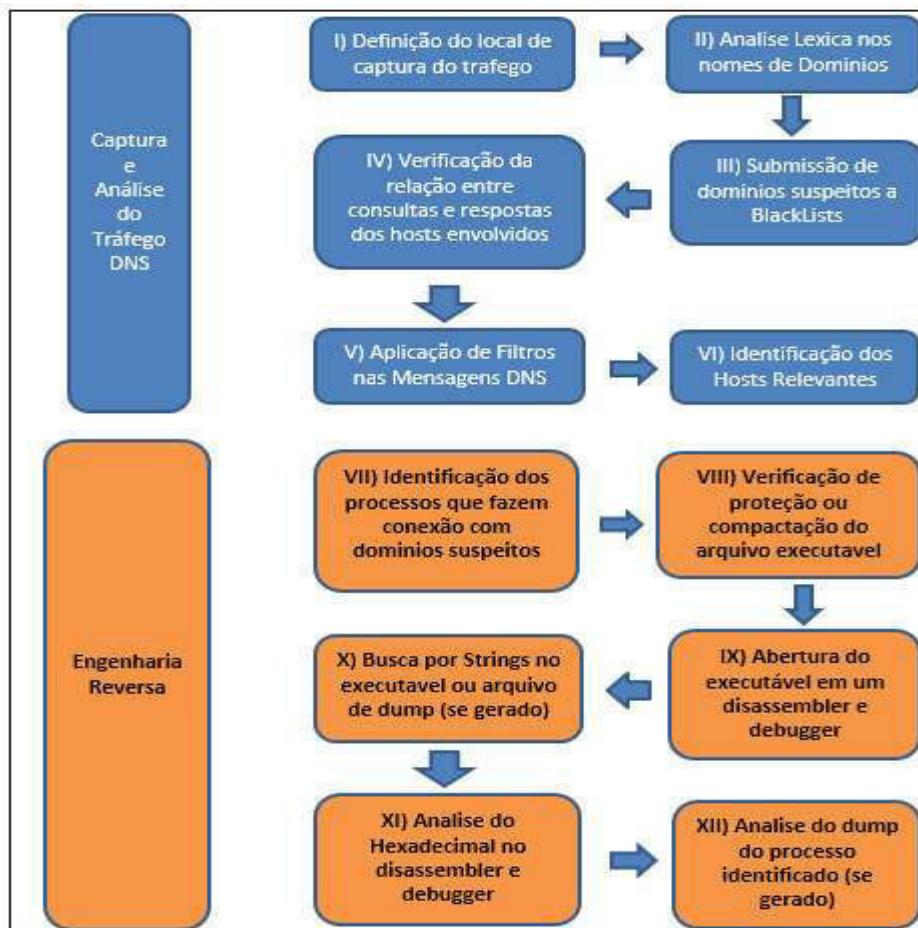
	Ass. de Rede	Eng. Reversa	Rast. de Pacote	Met. Hibrido	Trafego Rede	DNS
Ceron 2010	X					
Laufer 2005			X			
Cunha 2011		X		X	X	
Hossain 2010					X	X
Proposto		X		X	X	X

Tabela 1 – Comparativo entre o método proposto e os trabalhos relacionados

O presente artigo tem como objetivo, definir um método híbrido composto pela análise do tráfego de rede e a engenharia reversa, para detecção de *botnets* que utilizam o serviço de DNS para se proliferar e controlar seus *bots*. A intenção deste trabalho é ser um guia para a detecção com êxito de *botnets* que especificamente utilizam o serviço de DNS. O artigo está organizado da seguinte forma: A seção 2 apresenta o detalhamento do método proposto, a seção 3 apresenta um estudo de caso e os resultados obtidos, e a seção 4 apresenta as conclusões.

2. Método Proposto

O método é dividido em duas partes: a primeira parte trata da captura e análise do tráfego DNS. A segunda parte trata da engenharia reversa feita no código malicioso. A Figura 1 exibe o diagrama da ordem de execução e a própria estrutura do método. Cabe ressaltar que outras ferramentas também podem ser utilizadas para a execução do método, alem das citadas, desde que cumpram o propósito definido em cada etapa.

**Figura 1 - Diagrama Roteiro e Estrutura do Método Proposto**

I) Definição do local de captura do tráfego

O host necessita fazer a interligação entre a rede interna e a rede externa, sendo assim geralmente atribuído a um Proxy, Firewall, Gateway ou Roteador executando algum sistema operacional derivado de Unix. Este host pode possuir uma ou mais placas de rede.

II) Análise Léxica nos nomes de Domínios

A análise léxica é o processo de verificação da formação dos nomes de domínios, isso implica em perceber se os caracteres ou símbolos que formam o nome de domínio não estão fora do padrão de nomes comprehensíveis.

III) Submissão de domínios suspeitos a *BlackLists*

Sites descritos como *blacklists* possuem softwares que fazem a verificação sobre o domínio, e-mails, endereços de IP que foram previamente denunciados como disseminadores de mensagens consideradas SPAM.

IV) Verificação da relação entre consultas e respostas dos hosts envolvidos

O entendimento e classificação de comportamentos através do tráfego DNS podem ser obtidos por dois métodos básicos: A relação de consultas entre os hosts, e como as consultas DNS são realizadas, já que o registro de recursos denota o objetivo da consulta.

V) Aplicação de Filtros nas Mensagens DNS

Quanto ao como as consultas DNS são realizadas, devem ser levados em consideração: Estações que enviam consultas mal formadas, respostas que possuem erros de nomes verificando o MXDOMAIN, e uma grande quantidade de mensagens de erro geradas com um espaço de tempo reduzido.

VI) Identificação dos Hosts Relevantes

Quando um hosts é reincidente em quase todas as verificações e filtros executados.

VII) Identificação dos processos que fazem conexão com domínios suspeitos

Após identificar os hosts relevantes, devem ser verificados os status das conexões ativas nestes hosts e identificar o PID do processo que esta em execução e fazendo a conexão.

VIII) Verificação de proteção ou compactação do arquivo executável

Grande parte dos *malwares* sofrem alterações após a compilação. Eles são protegidos ou compactados, para que não sejam detectados por aplicações anti-virus.

IX) Abertura do executável em um *disassembler* e *debugger*

É preciso desmontar o programa e verificar se o domínio ou endereço IP está contido no executável, ou em um arquivo de *dump* gerado a partir de um processo.

X) Busca por *Strings* no executável ou arquivo de *dump* (se gerado)

As *strings* possuem o endereço IP do servidor, ou o domínio ao qual o software deve se conectar, elas também revelam muito sobre o comportamento do software.

XI) Analise do Hexadecimal no *disassembler* e *debugger*

As vezes pode acontecer de as *strings* não revelarem tudo sobre o programa, tendo a alternativa de verificação do hexadecimal gerado.

XII) Analise do *dump* do processo identificado (se gerado)

Este passo é opcional mas importante, visto que o *dump* contém todo o contexto no qual o processo sofreu o despejo.

3. Estudo de Caso – Detecção de uma *botnet*

Neste estudo de caso é apresentado a detecção de uma *botnet* através da análise do tráfego de DNS. Para demonstrar este estudo de caso, foram configurados um servidor DNS e um servidor de comando e controle (C&C). O domínio criado foi chamado de “botnetstolpe.net” fazendo-se com que o servidor DNS fosse autoritativo sobre este domínio. O ambiente no qual o método foi aplicado é de produção com tráfego de dados reais, composto por aproximadamente 60 dispositivos conectados à internet, dentre eles estações de trabalho, notebooks, servidores e dispositivos móveis. Na configuração do servidor de comando e controle (C&C) utilizou-se o programa DarkDDoS instalado em um sistema Windows 7.

O inicio da aplicação do método começa com a captura do tráfego DNS, sendo executado apartir de um terminal de comando em ambiente Unix, no gateway principal da rede:

tcpdump -i eth1 src port 53 or dst port 53 -w capturaDNS-Dia_Hora.pcap

Posteriormente foi efetuado a analise léxica do nome de domínio. Um endereço IP pode ser identificado como banda larga caso o nome de domínio seja composto por palavras como cpe, vivax, adsl, modem, virtual, e isto serve como um critério básico de exclusão para estes domínios, quando há uma desconfiança na formação dos nomes. O *sniffer Wireshark* permite gerar um arquivo contendo a tradução dos nomes de domínios para endereços IPs, através do resolvedor de endereços na guia de estatísticas. Os dominios listados com o Wireshark foram inseridos em um arquivo de texto e comparados através de um comando linux, com uma lista de dominios baixados do site [<http://urlblacklist.com>], que divide os nomes listados em categorias, como por exemplo, *hacker, mail, spyware, proxy, ddos* entre outros. O comando utilizado foi: ***grep -xf dominios.txt dominiosBaixados.txt***, que retornou em tela no terminal, apenas os nomes de domínios que faziam parte das *blacklists* categorizadas.

Posteriormente foram verificados se cada nome retornado fazia parte de outras *blacklists*, através do site [<http://mxtoolbox.com/blacklists.aspx>] que para alguns domínio retornou positivo. Após obter os domínios classificados como ameaça, efetuou-se a verificação no *dump* de rede, de quais endereços IPs consultaram os domínios listados. O comando utilizado no *Wireshark* para tal verificação foi ***dns.qry.name contains "nomeDoDominio"***.

O próximo passo é entender o padrão de comunicação entre os hosts identificados como possíveis *bots*, sendo necessário relacionar o total de consultas realizadas, com o total de respostas, isto é possível através da visualização de conversações efetuadas entre os hosts, que tambem são mostradas na guia de estatísticas do Wireshark. Fazendo a relação entre as consultas, foi examinado o formato das mesmas, verificando-se a integridade e conteúdo de cada uma. A primeira tarefa foi a verificação da integridade das consultas, através das flags das mensagens DNS, como: “***dns.flags.rcode == 1***” e “***dns.flags.rcode ==3***” no filtro do *Wireshark*, é possível obter as consultas que retornaram com o status de MXDOMAIN (*No Such Name*), que podem corresponder a servidores de comando e controle. Outro modo é através da análise visual do *dump* de rede, procurando por diferentes endereços IPs consultando um único endereço IP, em um intervalo de tempo muito curto. Os hosts que apresentarem estes comportamentos serão tratados como “hosts relevantes”, pois apresentam o comportamento de um servidor que envia um comando para que os clientes(*bots*) respondam a um comando, todos ao mesmo tempo. De posse dos hosts relevantes e os domínios suspeitos, efetuou-se nova análise no

trafego DNS em todos arquivos de captura, fazendo-se a correlação entre quais os hosts que acessaram quais domínios, assim como a frequência em que acessavam. O comando para a verificação dos domínios em cada arquivo de captura foi: ***dnsqry.name contains "nomeDoDominio"***.

Foi necessário relacionar o total de consultas realizadas com o total de respostas, procurando por diferentes endereços IPs consultando um único endereço IP, em um intervalo de tempo muito curto. Este fato pode ser observado em dois momentos no trafego DNS capturado em um dos dias, onde alguns endereços IPs consultam o servidor DNS no mesmo segundo. No primeiro momento as 10:03:03 acontece a consulta de diferentes endereços procurando pelo servidor do domínio “botnetstolpe.net”.

As Figuras 2 e 3 exibem o momento do ataque, com uma diferença de 24 segundos entre a consulta dos *bots* pelo domínio, e o ataque propriamente dito.

Executou-se um ataque de DoS (Negação de Serviço) em um roteador na rede. No segundo momento as 11:35:01 acontece o segundo ataque, e o trafego demonstra as consultas efetuadas pelos *bots*.

10:03:03	192.168.1.97	192.168.1.210	DNS	86 Standard query 0x648a A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.97	DNS	150 Standard query response 0x648a A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.70	192.168.1.210	DNS	86 Standard query 0x16f8 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.70	DNS	150 Standard query response 0x16f8 A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.63	192.168.1.210	DNS	86 Standard query 0x98b4 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.63	DNS	150 Standard query response 0x98b4 A serverbot.botnetstolpe.net A 192.168.1.125
10:03:03	192.168.1.95	192.168.1.210	DNS	86 Standard query 0x81f2 A serverbot.botnetstolpe.net
10:03:03	192.168.1.210	192.168.1.95	DNS	150 Standard query response 0x81f2 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.66	192.168.1.210	DNS	86 Standard query 0x644a A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.66	DNS	150 Standard query response 0x644a A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.70	192.168.1.210	DNS	86 Standard query 0x7ac9 A serverbot.botnetstolpe.net
11:35:01	192.168.1.95	192.168.1.210	DNS	86 Standard query 0x4b45 A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.70	DNS	150 Standard query response 0x7ac9 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.210	192.168.1.95	DNS	150 Standard query response 0x4b45 A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.97	192.168.1.210	DNS	86 Standard query 0x055e A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.97	DNS	150 Standard query response 0x055e A serverbot.botnetstolpe.net A 192.168.1.125
11:35:01	192.168.1.63	192.168.1.210	DNS	86 Standard query 0x1102 A serverbot.botnetstolpe.net
11:35:01	192.168.1.210	192.168.1.63	DNS	150 Standard query response 0x1102 A serverbot.botnetstolpe.net A 192.168.1.125

Figura 2 - Trafego suspeito no servidor DNS

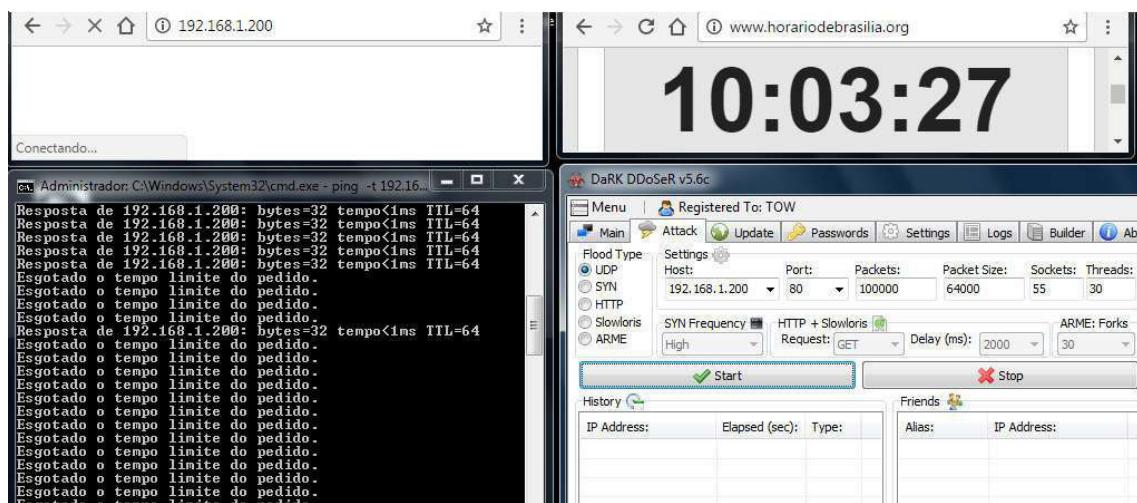


Figura 3 - Ataque de negação de serviço (DoS) no roteador

Para a identificar os hosts relevantes, efetuou-se o cálculo da quantidade de vezes que um determinado endereço IP é exibido em um dia e em um filtro. Ao todo obteve-se 8 hosts que apresentaram alguma característica inerente aos filtros. E neste caso o fator que mais contribuiu para a relevância do host ser considerado suspeito, foi a sua inserção no grupo dos hosts acessando ao mesmo tempo um único IP.

Ao inicializar o sistema operacional dos hosts, que neste momento são considerados bots verificou-se o status das conexões ativas através do comando “netstat –o” para identificar o PID do processo que está fazendo a conexão. A Figura 4 mostra que há uma conexão

de um processo chamado botClient.exe do host 192.168.1.70, na porta 5555 de um servidor chamado SERVERBOT.

```

Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>netstat -o

Conexões ativas

  Proto  Endereço local          Endereço externo        Estado      PID
  TCP    192.168.1.70:49158      SERVERBOT:5555       ESTABLISHED  2808

C:\Windows\system32>

```

Proto	Endereço local	Endereço externo	Estado	PID
TCP	192.168.1.70:49158	SERVERBOT:5555	ESTABLISHED	2808

Figura 4 - Conexão de um bot com o Servidor

Após obter o arquivo botClient.exe o mesmo foi submetido ao programa EXEinfoPE para verificação de possível ofuscação no código. Entretanto foi possível observar que o mesmo não possuia nenhuma proteção, e que este arquivo foi desenvolvido na linguagem de programação Delphi.

Ao abrir o executável no *disassembler* IDA PRO e no *debugger* OllyDBG, não foram encontradas nenhuma referência ao servidor, como nome do servidor, porta ou endereço IP, tanto na busca por *strings* quanto no hexadecimal. Apenas foram encontradas referências a classes de programação, chaves do registro do Windows, e *System Calls* do sistema operacional.

Entretanto ao fazer o *dump* de memória com o processo em execução, gerou-se um arquivo chamado botClient.DMP, e este foi submetido ao IDA PRO para análise.

Ao verificar este arquivo foi possível constatar o endereço do servidor de comando e controle chamado “serverbot.botnetstolpe.net”, ao qual o *bot* conectava-se, como mostra a Figura 5.

```

seg000:000B7EA0 00 02 00 18 00 00 00 00 00 00 00 00 00 00 00 73  .-+-----+...S.
seg000:000B7EB0 00 65 00 72 00 76 00 65 00 72 00 62 00 6F 00 74  .e.r.v.e.r.b.o.t
seg000:000B7EC0 00 2E 00 62 00 6F 00 74 00 6E 00 65 00 74 00 73  ...b.o.t.n.e.t.s
seg000:000B7ED0 00 74 00 6F 00 6C 00 70 00 65 00 2E 00 6E 00 65  .t.o.l.p.e....n.e
seg000:000B7EE0 00 74 00 00 00 01 00 00 00 00 00 00 00 00 50 00  .t.... .....P,
seg000:000B7FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Figura 5 - Domínio encontrado no Arquivo de Despejo

Ou seja, neste caso a detecção do domínio que liga o *bot* ao servidor de comando e controle, só foi possível através do arquivo de despejo de memória, executado sobre o processo em execução. Isso demonstra que, mesmo sendo parte opcional do método proposto, são obtidos resultados satisfatórios.

4. Conclusão

O presente trabalho teve por objetivo descrever e aplicar um método híbrido composto pela análise do tráfego DNS em conjunto com a engenharia reversa de código malicioso, para detecção de botnets.

Após a aplicação do método, o mesmo mostrou-se satisfatório, pois além do objetivo principal que é a detecção de *botnets*, ele também consegue distinguir anomalias no tráfego de DNS, quando estas são causadas por algum tipo de *malware*.

A contribuição acadêmica deste trabalho é importante por demonstrar que em algum momento entre a comunicação de programas maliciosos, com um servidor ou controlador de *malwares*, é possível detectar comportamentos estranhos ou anômalos para mitigar qualquer espécie de ameaça que utiliza o protocolo DNS contra uma rede de computadores.

Referências Bibliográficas

- [Araújo et al.2010] J. M. Araújo Filho (2010, pt. 2) Ciberterrorismo e Cibercrime: o Brasil está preparado?
- [Binsalleh et al. 2010] Binsalleh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., and Wang, L. (2010). On the analysis of the zeus botnet crimeware toolkit. In Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on, pages 31–38.
- [Ceron J., Granville L., Tarouco L] João Marcelo Ceron, Lisandro Zambenedetti Granville, Liane Margarida Rockenbach Tarouco (2010) Uma Arquitetura Baseada em Assinaturas para Mitigação de Botnets.
- [Ceron et al.2010] Ceron, João Marcelo (2010) Arquitetura Distribuída e Automatizada para mitigação de botnets baseada em análise dinâmica de malwares.
- [Cunha Neto et al.2011] Cunha Neto, Raimundo Pereira da (2011) Sistema de Detecção de intrusos em ataques oriundos de botnets utilizando método de detecção híbrido.- São Luis PPGEE.
- [Craig 2007] Craig A. Schiller, Botnets - The Killer Web App, Singress 2007.
- [Davis et al. 2008] Davis, C., Fernandez, J., Neville, S., and McHugh, J. (2008). Sybil attacks as a mitigation strategy against the storm botnet. In Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on, pages 32–40.
- [Egele et al. 2008] Egele, M., Scholte, T., Kirda, E., and Kruegel, C. (2008). A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv., 44(2):6:1–6:42.
- [Ferreira 2013] Ferreira Pedro, Detecção de Botnets, IPB 2013.
- [Hossain et al. 2010] Soraya Sybele Hossain, Detecção de aplicações envio de Spam através da mineração do tráfego DNS(2010).
- [Kaio 2014] Kaio Rafael, Identificação e Caracterização de Comportamentos Suspeitos Através da Análise do Tráfego DNS. SBSEG 2014.
- [LAUFER et al.2005] LAUFER RAFAEL PINAUD Rastreamento de Pacotes IP contra Ataques de Negação de Serviço [Rio de Janeiro] 2005 XIII, 93 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)
- [McCarty 2003b] McCarty, B. (2003b). Botnets: Big and bigger. IEEE Security and Privacy, 1(4):87–90. cited By (since 1996)41.
- [Morimoto 2013] Morimoto, Carlos Eduardo. Servidores Linux – Guia Prático, 1º ed. SulEditores 2013.
- [Mota Filho 2013] Mota Filho, João Eriberto, Análise de Trafego em redes TCP/IP. 10.ed.Novatec 2013.