Honeypot de baixa interação como ferramenta para detecção de tráfego com propagação de botnets.

Dino Amaral, Laerte Peotta

Universidade de Brasília - Departamento de Engenharia Elétrica
Faculdade de Tecnologia - Campus Darcy Ribeiro
Bairro Asa Norte. CEP: 70910900 - Brasília - DF

E-mails: dinomacedo@yahoo.com.br e peotta@unb.br

Abstract— Something bigger than we imagine is happening in the Internet, many hosts are serving the hackers and may be used at any moment with a single command that launch many type of attacks. In this paper, it will be shown a method to detect botnets using honeypots.

Keywords— Botnets, Honeypots, Worms, Security Information, Progamming C

Resumo— Algo grandioso está ocorrendo atualmente na Internet, milhares de máquinas estão totalmente a mercê de criminosos e podem ser utilizadas a qualquer momento com um único comando para uma muitos tipos de ataques. Neste artigo será descrito um método de se detectar botnets utilizando honeypots.

Palavras-chave— Botnets, Honeypots, Worms, Segurança da Informação, Programação em C

1 Introdução

Conseguir encontrar padrões de comportamento nos ataques às estações vulneráveis consiste em uma tarefa árdua para os analistas de segurança da informação, para tal existem diversos dispositivos em um perímetro de defesa como *firewalls*, IDS (*Intrusion Detection Systems*), IPS (*Intrusion Prevetion System*), ADS (*Anomaly Detection System*), Servidores *Proxies* entre outros. Deixando de ser um mito para pequenas empresas e usuários domésticos, esses dispositivos são amplamente usados, ora optando por utilizar software livre ora com soluções adequadas às demandas existentes em seus respectivos ambientes. O uso de *honeypots* tem se apresentado eficaz na detecção das tendências de ataques que permeiam a Internet [1], e fornecido uma fonte inesgotável para os pesquisadores, que através de seus *logs* mostram a anatomia de ataques ocorridos. Com o intuito de ganhar poderio computacional e com conceitos de computação distribuída consolidadas, o uso de *botnet* (*robot network*) tem sido alvo para criminosos cibernéticos, pois a possibilidade de ter algumas centenas de computadores sob controle, e que respondam a único comando de maneira eficaz, torna-se extremamente atraente para estes criminosos. Dentre os motivos inicias que permeiam a formação de uma botnet, podemos enumerar : estações, quando infectadas, estarão aptas a enviar e-mails indesejados (*spam*) em quantidade expressiva e ataques de negação de serviços distribuídos (*DDoS*). Neste artigo serão apresentadas questões que envolvem a formação de *botnets* utilizando ferramentas como honeyd [2], um honeypot de baixa interação[3] amplamente usado, criado para capturar tráfegos maliciosos.

Em um *honeypot* de baixa interatividade são instaladas ferramentas para emular sistemas operacionais e serviços com os quais os atacantes irão interagir. Desta forma, o sistema operacional real deste tipo de *honeypot* deve ser instalado e configurado de modo seguro, para minimizar o risco de comprometimento[16].

Este Artigo está estruturado da seguinte maneira: A seção 2 trata dos conceitos de botnets e modo de operação. A seção 3 trata da motivação de utilização. A seção 4 aborda conceitos e métodos de utilização de honeypots de baixa interação. Dois estudos de caso são demonstrados na seção 5. Conclusões e trabalhos futuros são discutidos na seção 6.

2 Botnet – Máquinas "zumbis"

O termo *botnet* é uma referência a utilização de robôs, que se utiliza de códigos maliciosos na tentativa de transformar um computador "normal" em um zumbi (termo utilizado para máquinas comprometidas que fazem parte de uma *botnet*). Esses zumbis passam a ser controlados a distância, independente da vontade do usuário local.

Segundo o FBI (*Federal Bureau of Investigation*) e o Departamento de Justiça dos Estados Unidos somente nos Estados Unidos foram identificados mais de um milhão de máquinas zumbis. Com essas informações, está claro o grande potencial de danos que essas redes podem impor não somente a infra-estrutura da Internet, mas também a segurança nacional e a economia global.

2.1 *Modo de operação de uma botnet convencional*

No primeiro passo o atacante explora alguma vulnerabilidade da máquina alvo e envia o *bot*, transformando a vítima em um zumbi. (figura1)

Figura 1: Envio do *bot*

No passo seguinte, o atacante disponibiliza os recursos computacionais da máquina infectada. Neste caso específico, temos um caso clássico de envio de *spam*, onde o *spammer* paga por um determinado tempo o uso destas máquinas *zumbis* para envio de *e-mails* não solicitados. Todo o controle é feito por parte do atacante que faz o processamento através de um servidor *IRC (Internet Relay Chat)*, onde as vítimas ao se conectarem a Internet se conectam ao servidor *IRC* diretamente. (Figura 2)

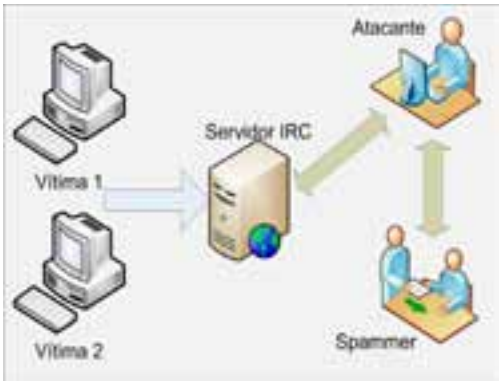

Figura 2: Acordo de utilização de *botnet*

Depois do acordo feito, que geralmente envolve valores financeiros o atacante libera uma parte da *botnet* que irá prover o serviço ao *spammer*. Inicia-se a fase crítica de envio de *e-mails*, esta etapa do processo reside no fato de cada vítima receber, do servidor IRC, os comandos e as mensagens que deverão ser enviadas, esta troca de mensagens e comandos acontece sem o conhecimento da vítima. (Figura 3). O que pode chamar a atenção de um usuário mais meticuloso é a perda de performance de seus recursos computacionais.

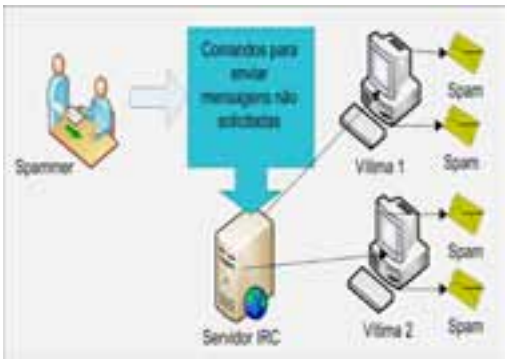

Figura 3: Utilização da *Botnet.*

3 Motivação de utilização de botnets

Mediante a coleta de informações nos *honeypots* instalados para esta pesquisa, deparou-se com a necessidade de dissecar os *logs* gerando uma retroalimentação na política de segurança. Uma *botnet* não é nada mais que uma ferramenta e existem muitas razões para usá-las. Os motivos mais comuns são as utilizações criminosas visando ganhos monetários ou para propósitos destrutivos. Baseados no tipo de dados que foram capturados, as possibilidades de usar uma *botnet* podem ser atribuídas de diversas maneiras. Pode-se afirmar que existe uma lista extensa para uso dessas redes e as que estão expostas neste artigo consistem em razões mais comuns e as que foram detectadas nesta pesquisa, e são elas:

- Ataques distribuídos de Negação de Serviço: De maneira geral as *botnets* são usadas para ataques *DdoS (Distributed Denial of Service)*, que consiste no comprometimento de estações ou infra-estrutura de redes que pode causar a paralisação de algum serviço para os usuários, seja através do consumo excessivo de banda ou de sobrecarga nos recursos computacionais na estação da vítima. Os recursos da estação são consumidos através do envio de pacotes em uma alta taxa com tamanhos diferenciados. As *botnets* analisadas incluem diferentes possibilidades de causar um ataque de *DDoS*, na maioria das vezes com o envio de pacotes *TCP* (*Transmission Control Protocol*) sinalizando o início de conexão com o bit *SYN* marcado ou com uma inundação de pacotes *UDP (User Datagram Protocol)*.
- *Spam*: Em [4] é mostrado as estatísticas do envio de *spam* [5] pelo Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.Br), algumas *botnets* oferecem a possibilidade de abrir uma conexão com *SOCKS proxy* na estação comprometida. Com este serviço habilitado, a máquina poderá ser usada para tarefas como o envio de *spam*. Com a ajuda de

botnet e alguns milhares de *bots*, é possível o envio de *spam* de forma automatizada e em quantidade massiva. As estatísticas mostram que o sistema operacional *Windows*, sem atualizações, é o maior alvo de *spammers*.

- Capturar tráfego: As máquinas comprometidas que fazem parte de uma *botnet* podem ser usadas para capturar tráfego que são transmitidos em claro, sem nenhum tipo de criptografia. *Sniffers*, como *tcpdump* [6] e *Wireshark* [7], são usados para capturar informações como nome de usuários e senhas. Capturar tráfego de uma estação, que está sendo usada por mais de uma *botnet*, possibilita que os dados pertinentes de uma das *botnets* e assim que a mesma seja roubada do "concorrente", prática muito comum no mundo dos criminosos digitais.
- *Keylogging*: Se a máquina comprometida usar algum tipo de criptografia nos canais de comunicação, capturar tráfego com *sniffers* torna-se uma tarefa inválida, pois seria necessário a descoberta das chaves utilizadas para cifrar o tráfego. Com a ajuda de um *keylogger*, a tarefa de capturar dados digitados pelo usuário na estação comprometida torna-se extremamente fácil. Geralmente, os dados digitados são enviados a um e-mail, através da porta 25 (*SMTP - Simple Mail Transfer Protocol*), e em seguida é realizada uma procura por palavras como de sites de bancos ou sites de pagamentos, como *PayPal*.
- Instalação de propagandas e objetos de ajuda nos navegadores: *Botnets* podem ser usadas para ganhos financeiros. Esta tarefa é executada com a criação de um site com algumas propagandas sendo que o operador negocia com as empresas que hospedam páginas que pagam a cada clique em seus "*pop-ups*" de propagandas. Com a ajuda de uma *botnet*, estes cliques podem ser automatizados e rapidamente alguns milhares de *bots* clicam nos *pop-ups*, o processo pode ser melhorado por parte do atacante, configurando o navegador da vítima para clicar nos *pop-ups* de propaganda a cada vez que iniciar o seu próprio navegador.
- Propagação de códigos maliciosos: As *botnets* são usadas para capturar mais *bots* que irão se integrar a uma *botnet*, e assim aumentar o número de estações comprometidas. A maioria dos *bots* implementa mecanismos de *downloads* ou executam um arquivo via *HTTP (HyperText Transfer Protocol*) ou *FTP (File Transfer Protocol*). Imaginando uma *botnet* com 5.000 estações usadas como base de ataques para disseminar um vírus por e-mail. Um exemplo desta técnica foi o *worm Witty* [14], que atacou o protocolo usado pelo ICQ um programa para comunicação instantânea.

Os motivos descritos neste artigo não são os únicos, pode-se citar:

a) Manipulação em *sites* de apostas, visto que as votações são rastreadas por endereço IP;
b) Ataques as canais de *chats*, algo similar aos ataques de negação de serviço, dentre outros.

Mapear a motivação das *botnets* é uma fonte inesgotável para os pesquisadores. A contínua pesquisa por estes tópicos tem achado nos *honeypots* um ótimo aliado, pelos conceitos que os cercam torna-se mais fácil de analisar este tipo de comportamento. Existem diversas razões pelas quais os *honeypots* podem auxiliar na detecção de *botnets*, ou tentativas de formação delas:

1 Com o auxilio dos *honeypots*, é possível encontrar informações importantes, como endereço IP do servidor ou *nickname* da *bot*, que possibilita observar a *botnet*, podendo inclusive, coletar os arquivos-fontes dos programas, que tentam se instalar na estação a ser comprometida.
2 Podem-se monitorar os comandos executados assim como toda a comunicação entre o atacante e a vitima, permitindo descobrir as técnicas usadas pelo atacante e suas motivações.
3 Os arquivos de *log* constituem uma fonte para encontrar tendências dos ataques, pois se considera que todo o tráfego destinado ao *honeypot* ou é um ataque ou é tráfego malicioso. Nesses *logs* encontram-se subsídios suficientes para iniciar um contra-ataque à formação de *botnets*.

4 Tráfego de *Botnets* em um *Honeypot*

A principal função de um *honeypot* é atrair um eventual atacante, monitorando e registrando todas as suas ações, buscando obter o modo de operação bem como ferramentas utilizadas tanto para comprometer o *honeypot*, quanto as utilizadas para busca de novos *hosts* vulneráveis. Outra função muito importante é a descoberta de novas vulnerabilidades que estão sendo exploradas sem conhecimento de órgãos de segurança da Internet.
Pode-se saber com antecedência as tendências de novos ataques e a difusão de *worms* pela rede.
Nesta pesquisa foi utilizado um *honeypot* de baixa interatividade buscando filtrar tráfego e

tentativa de criação ou mesmo de aumento de *botnets*.

A escolha pela utilização de um *honeypot* de baixa interação (figura 4) é devida principalmente pela facilidade de instalação e manutenção, por incorrer em menores riscos, pois os serviços são virtualizados ou emulados, outro fator importante é a de que não é preciso incorporar mecanismos de contenção, pois todo o trafego já é contido na própria ferramenta.

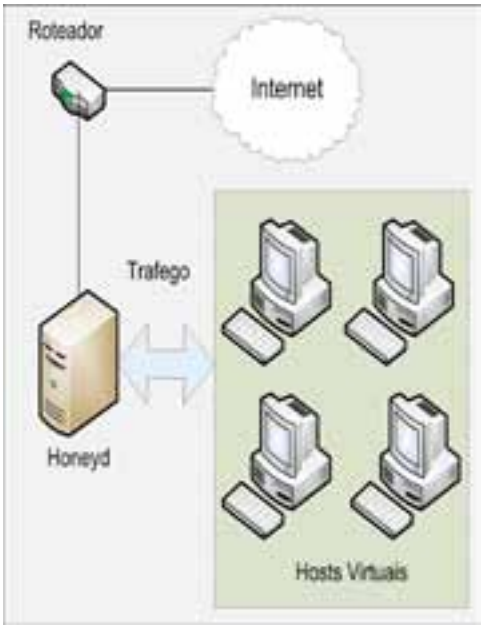

Figura 4: Topologia de um *honeypot* de baixa interatividade

5 Estudos de casos

Os casos descritos nesta seção mostram o comportamento dos atacantes, quando os mesmos obtêm acesso aos sistemas. No *honeypot* montado para este estudo, disponibilizou-se uma conexão via SSH (*Secure Shell*) com um conjunto fraco de nome de usuário e senha. Configurou-se o *honeypot* para capturar os comandos executados, fornecendo respostas pré estabelecidas nos *daemons* do *honeyd*. Os resultados dessas interações são enviados para um arquivo de log, que permite decifrar as motivações de uma invasão à um desses *hosts* virtuais. Por questões do compromisso firmado com o Consórcio Brasileiro de *Honeypots*, administrado pelo CERT.br, os endereços dos hosts virtuais não serão divulgados. É importante ressaltar a possibilidade de alteração dos arquivos de *logs*, caso o atacante consiga acesso ao *host* que hospeda os *logs* da aplicação.

a) Caso 1

Na figura 5 seguem os *logs* de um ataque a um dos *hosts* do *honeypot*. O atacante, inicialmente, consegue acesso através da porta TCP 22, na qual existe um "*daemon*" configurado para responder com usuário e senha, considerados fracos, neste caso usuário=root e senha=root. Ao obter acesso, o arquivo de *log* fornece dados relevantes do atacante, como endereço IP de origem (87.258.168.110) e o Sistema Operacional utilizado pelo *host* remoto (Windows XP SP1).

```
2007-06-03       12:15:39       +0000:
fakersh[15121]:    HONEYD_DST_PORT=22,
HONEYD_IP_DST=201.XX.XX.XX,
HONEYD_IP_SRC=87.248.168.110,
HONEYD_PERSONALITY=Linux Kernel 2.4.3
SMP (RedHat), HONEYD_REMOTE_OS=Windows
XP     SP1,     HONEYD_SRC_PORT=1816,
LOGNAME=root,
PATH=/bin:/usr/bin:/usr/sbin,
TERM=xterm, USER=nobody
```

Figura 5: log de ataque ao *honeypot*

Com acesso máximo ao sistema, pois conseguiu efetuar *logon* como usuário ***root***, os *logs* (figura 6) mostram os comandos executados pelo atacante. O atacante procura saber o conteúdo do diretório, inclusive os arquivos ocultos com o comando "ls –a", o qual não obtém sucesso, então o atacante executa o comando "ls" sem parâmetro e obtém como resposta os arquivos fictícios do mrtg (*Multi Router Traffic Grapher*), uma ferramenta desenvolvida para monitorar o tráfego em *links* de rede.

```
2007-06-03       12:15:43       +0000:
fakersh[15121]: in: ls -a
2007-06-03       12:15:43       +0000:
fakersh[15121]:   out:   bash:   fork:
Resource temporarily unavailable
2007-06-03       12:15:45       +0000:
fakersh[15121]: in: ls
2007-06-03       12:15:45       +0000:
fakersh[15121]:     out:     mrtg.cfg
mrtg.cfg~ mrtg.ok
```

Figura 6: *Log* dos comandos executados

Após fazer o reconhecimento inicial do ambiente, o atacante inicia a tentativa de aumentar ou iniciar a formação de sua *botnet*, ao tentar efetuar o *download* do arquivo empacotado e compactado (figura 7), no formato *UNIX* (tar.gz) que contém os arquivos de configuração para a formação de uma *botnet*. Antes que fosse removido deste site (`http://Bnc-Irc.trei.ro/linux`), efetuou-se o *download* do arquivo que foi analisado como sendo os arquivos fontes de configuração da referida *botnet.*

```
2007-06-03       12:15:51       +0000:
fakersh[15121]: in: wget
2007-06-03       12:15:51       +0000:
fakersh[15121]: out:
wget: missing URL
Usage: wget [OPTION]... [URL]...
Try `wget --help' for more options.
2007-06-03       12:16:06       +0000:
fakersh[15121]: in: wget http://Bnc-
Irc.trei.ro/linux/bots.tar.gz
2007-06-03       12:16:08       +0000:
fakersh[15121]: out:
--01:32:01--              http://Bnc-
Irc.trei.ro/linux/bots.tar.gz
```

```
                    => `http://Bnc-
Irc.trei.ro/linux/bots.tar.gz'
Resolving                  http://Bnc-
Irc.trei.ro/linux/bots.tar.gz...
failed: no address associated with
name.
```

Figura 7: Tentativa de *download*

Os arquivos fontes possuem dados preciosos para uma análise. Ao extrair os arquivos, notou-se que a conexão na *botnet* aconteceria através de um canal de *chat*, os servidores que estariam aptos a receber este *host* estão listados na tabela 1.

Tabela 1: Relação de servidores IRC

Servidor	Porta
Amsterdam2.NL.EU.undernet.org	6669
graz.at.Eu.UnderNet.org	6670
Helsinki.FI.EU.Undernet.org	6666
Lelystad.NL.EU.UnderNet.Org	6668
Stockholm.SE.Eu.Undernet.org	6666
washington.dc.us.undernet.org	6667
geneva.ch.eu.undernet.org	6667
Ede.NL.EU.UnderNet.Org	7000
graz2.at.Eu.UnderNet.org	6670
London.UK.Eu.UnderNet.org	6667
Oslo2.NO.EU.undernet.org	7000
Miami.FL.US.Undernet.org	6667
mesa.az.us.undernet.org	6667
EU.UNDERNET.ORG	6667

O canal especificado para os *hosts* se conectarem é o "*#eggucs*" e o *nickname* é o "*ucs123"*, ao se conectarem neste canal com o *nickname* mencionado, o *botmaster*, o *host* que possui o controle sobre todos os hosts que estão conectados neste canal, está apto a enviar qualquer comando para os *hosts*. Os comandos podem ser enviados para a *botnet* de duas maneiras: a primeira é enviando um comando diretamente para a *botnet* e a segunda é enviando um tópico especial no canal que todos os hosts interpretam, como por exemplo: "***advscan lsass* 200 5 0 -b**", o qual envia um comando a *botnet* para espalhar uma conhecida vulnerabilidade da plataforma *Microsoft*, os *hosts* começam 200 sessões simultaneamente com um atraso de 5 segundos por tempo ilimitado, o parâmetro "-b" significa que os *hosts* alvos estão dentro da classe B.

b) Caso 2

Como no Caso 1, o usuário obteve acesso ao sistema como usuário root e senha root (figura 8), com endereço IP de origem 87.106.101.28 e sistema operacional do host do atacante é um *Linux* com *kernel* versão 2.6.

```
2007-06-07       14:51:37       +0000:
fakersh[6515]:     HONEYD_DST_PORT=22,
HONEYD_IP_DST=201.XX.XX.XX,
HONEYD_IP_SRC=87.106.101.28,
HONEYD_PERSONALITY=Linux Kernel 2.4.3
SMP (RedHat), HONEYD_REMOTE_OS=Linux
2.6      ,      HONEYD_SRC_PORT=41860,
LOGNAME=root,
PATH=/bin:/usr/bin:/usr/sbin,
TERM=xterm, USER=nobody
```

Figura 8: Acesso ao sistema

Em seguida, o atacante executa o comando "**w**" (abreviação de *who*), na qual mostra quem está atualmente conectado no computador (figura 9). Este comando, *who*, mostra os seguintes dados : os nomes de usuários que estão conectados no servidor, o terminal a qual está conectado e data da conexão. Em seguida, o atacante procura saber mais informações sobre o sistema operacional, versão do *kernel*, arquitetura do *host* com o comando "***uname -a***", neste caso o *daemon* está configurado para fornecer a seguinte resposta : "**Linux localhost 2.6.13.1 #1 Mon Sep 19 21:36:25 PST 2005 i586 GNU/Linux**". Antes de tentar baixar o software para a inclusão do *host* em sua *botnet*, o atacante executa o comando "**uptime**" para saber a quanto tempo o sistema está no ar.

```
2007-06-07       14:51:38       +0000:
fakersh[6515]: in: w
2007-06-07       14:51:38       +0000:
fakersh[6515]: out:
 11:29:34 up  3:17,  2 users,  load
average: 0,00, 0,00, 0,00
USER            TTY                FROM
LOGIN@   IDLE   JCPU   PCPU WHAT
root              tty2               -
08:12    4days  0.14s  0.13s -bash
root            ttyp0          localhost
10:11    0.00s  0.13s  0.00s w
2007-06-07       14:51:42       +0000:
fakersh[6515]: in: uname -a
2007-06-07       14:51:42       +0000:
fakersh[6515]:  out:  Linux  localhost
2.6.13.1 #1 Mon Sep 19 21:36:25 PST
2005 i586 GNU/Linux
2007-06-07       14:51:54       +0000:
fakersh[6515]: in: uptime
2007-06-07       14:51:54       +0000:
fakersh[6515]:  out:  bash:  uptime:
command not found
```

Figura 9: Executando comandos no *host* comprometido

A partir dos comandos executados, que significa um reconhecimento do ambiente na qual possui acesso, o atacante continua a sua trajetória com o intuito de aumentar o seu poderio computacional. Para dificultar sua detecção o atacante cria um diretório com o nome em branco, com o comando "*mkdir* **" "**", e em seguida cria um subdiretório com o nome de " ... ", o qual o mesmo deseja efetuar o *download* de sua aplicação. É interessante observar no fragmento do arquivo de *log* (tabela 2) que o atacante concatena os comandos com ";", e que após a tentativa de efetuar o *download* do arquivo-fonte, o mesmo tenta compilá-lo com o compilador gcc e depois tenta remover utilizando o comando "*rm -rf*", o que incorre em outra tentativa de dificultar a detecção do evento. Como aconteceu no Caso 1, efetuou-se o *download* do arquivo-fonte do site que é mostrado no arquivo

de *log* (tabela 2) e ao analisar os fontes percebeu-se que seria utilizado em ataques de Negação de Serviço Distribuídos. O *host* infectado é conectado nos servidores especificados que neste caso são: 194.109.20.90, 195.197.175.21, 195.68.221.221, 161.53.178.240, 69.16.172.34, 64.18.128.86 e 64.161.255.200 e aceita comandos através do canal especificado. A sintaxe é bem simples: !<*nick*> <*command*> . O atacante envia a mensagem para o canal que está definido no arquivo-fonte, onde <*nickname*> é o apelido que o *host* infectado se conecta no canal de IRC e o <*command*> é o comando a ser executado . Por exemplo, se o atacante deseja enviar para todos os clientes que possuam "*nickname*" que iniciam com N um comando, é digitar !N* <*command*>.
Existem vários comandos que podem ser executados remotamente:

a. *TSUNAMI* <target><secs> = A *PUSH+ACK flooder* envia uma quantidade excessiva de pacotes *TCP* com os *flags PUSH* e *ACK* marcados;
b. *PAN <target><port><secs>* = A *SYN flooder* envia uma quantidade excessiva de pacotes *TCP* com o *flag SYN* marcado, o que denota o pacote *TCP* com o incio de conexão;
c. *UDP <target> <port> <secs>* = A *UDP flooder* envia uma quantidade excessiva de pacotes *UDP*.

Convém afirmar que a intenção, neste caso, é consumir recursos da *host* da vítima o que é caracterizado por ataque de Negação de Serviço (Figura 10).

```
2007-06-07       14:51:57       +0000:
fakersh[6515]: in: wget
2007-06-07       14:51:57       +0000:
fakersh[6515]: out:
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
2007-06-07       14:52:07       +0000:
fakersh[6515]:   in:   cd   /tmp;mkdir
.bash_history;cd  .bash_history;  mkdir
" ";cd  "  "  ;mkdir  "  ...   ";cd  "  ...
";wget
http://dragoc.braindead.hu/kaiten.c;gc
c   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c
2007-06-07       14:52:07       +0000:
fakersh[6515]:  out:  cd:  /tmp;mkdir
.bash_history;cd .bash_history;mkdir "
";cd  "  "  ;mkdir  "  ...  ";cd  "  ...
";wget
http://dragoc.braindead.hu/kaiten.c;gc
c   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c: restricted
2007-06-07       14:52:20       +0000:
fakersh[6515]:        in:        wget
http://dragoc.braindead.hu/kaiten.c;gc
c   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c
```

```
2007-06-07       14:52:22       +0000:
fakersh[6515]: out:
--01:32:01--
http://dragoc.braindead.hu/kaiten.c;gc
c   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c
                                    =>
`http://dragoc.braindead.hu/kaiten.c;g
cc   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c'
Resolving
http://dragoc.braindead.hu/kaiten.c;gc
c   kaiten.c   -o   bash;./bash;rm   -rf
kaiten.c...    failed:    no    address
associated with name.
2007-06-07       14:52:30       +0000:
fakersh[6515]: in: cd /var/tmp
2007-06-07       14:52:30       +0000:
fakersh[6515]:   out:   cd:   /var/tmp:
restricted
2007-06-07       14:53:59       +0000:
fakersh[6515]: in: exit
```

Figura 10: *Logs* e subsídios para estudo de caso.

6 Conclusão e trabalhos futuros

Devido à facilidade e acesso a computadores e redes de alta velocidade as *botnets* tem um grande campo para se desenvolver. No entanto deve-se ter uma maior preocupação, não apenas pelo usuário, que deve tomar precauções, mas também das autoridades, pois devem criar dispositivos para detectar e neutralizar essas redes o mais rápido possível. Essas redes podem ser utilizadas para uma infinidade de ações que vão deste ataques de negação de serviços como o de envio indiscriminado de mensagens não solicitadas.
Autoridades de diversos países se mostram preocupadas e buscam informações sobre como localizar e combater essas redes. Em trabalhos futuros pretende-se montar uma linha de combate juntamente com as autoridades, permitindo disponibilizar as informações coletadas a fim de localizar e neutralizar essas redes diretamente em *backbones* ou filtros. A ameaça é real e pode levar a prejuízos não somente financeiros, mas dependendo do ataque e local, a perdas de vidas. Uma pequena demonstração foi o ataque aos meios de comunicação sofrido pela Estônia, onde os atacantes, através de técnicas de negação de serviços, tirou do ar os principais sites do governo, afetando inclusive o acesso à Internet do país, gerando além de prejuízos financeiros uma crise diplomática com a Rússia [15].

7 Referências Bibliográficas

[1] Estudo de taxonomia de ataques e atacantes em um honeypot de alta interação Peotta, Laerte e Amaral, Dino. ICCyber 2006.
[2] www.honeyd.org acessado em 01/07/2007
[3] www.honeynet.org.br acessado em 01/07/2007
[4] www.cert.br/stats/spam/ acessado em 01/07/2007
[5] http://pt.wikipedia.org/wiki/Spam acessado em 01/07/2007

[6] www.tcpdump.org acessado em 01/07/2007
[7] www.wireshark.com acessado em 01/07/2007
[8] The ZombieRoundup: Understanding, Detecting, and Disrupting Botnets Evan Cooke - Farnam Jahanian, Danny McPherson Electrical Engineering and Computer Science Department Arbor Networks University of Michigan. 2005
[9] A Proposal of Metrics for Botnet Detection Based on Its Cooperative Behavior. Mitsuaki Akiyama, Takanori Kawamoto, Masayoshi Shimamura Teruaki Yokoyama, Youki Kadobayashi e Suguru Yamaguchi, Nara Institute of Science and Technology, Japan , 2007 International Symposium on Applications and the Internet Workshops (SAINTW'07).
[10] Honeypot Aware Advanced Botnet Construction and Maintenance. Cliff C. Zou, University of Central Florida e Ryan Cunningham, University of Central Florida. International Conference on Dependable Systems and Networks (DSN'06) pp. 199-208.
[11] Know Your Enemy: Learning about Security Threats (Livro). Honeynet Project The (Author). Addison-Wesley Professional; 2 edition (2004). ISBN-13: 978-0321166463.
[12] Virtual Honeypots: From Botnet Tracking to Intrusion Detection (Livro). Niels Provos e Thorsten Holz. Addison-Wesley Professional. (2007). ISBN-13: 978-0321336323.
[13] Botnets: The Killer Web Applications (Livro). Craig Schiller e Jim Binkley. Syngress (2007). ISBN-13: 978-1597491358.
[14] Analysis of the Witty Worm. Bruce Schneier (2005).
[15] Cyberattack in Estonia “what it really means”. Jose Nazario. Arbor Networks (2007)
[16] Honeypots e Honeynets: Definições e Aplicações. Cristine Hoepers, Klaus Steding-Jessen e Marcelo H. P. C. Chaves, 2007