

# Invisivelmente

A proteção de redes contra invasões não precisa exigir configurações mirabolantes ou softwares complexos. Com o HLBR, nada muda na configuração de rede, mas a segurança aumenta sensivelmente.

por Rogerio Ferreira

É bastante comum a situação em que o chefe pede ao administrador de rede que aumente a segurança de um servidor web Microsoft. É fácil imaginar que muitos sugeririam a adoção de uma solução de Código Aberto, como *LAMP* ou *Python*, *Zope* e *Plone*. Entretanto, nem sempre isso é viável, especialmente se houver a exigência de conservação do legado.

Nesses casos, após aplicar todos os remendos de segurança, uma alternativa muito interessante é implementar um sistema de prevenção de intrusão (*IPS*, na sigla em inglês).

Soluções de *IPS* comerciais costumam ter alto custo, o que geralmente já é suficiente para justificar a busca de alternativas gratuitas. Este artigo descreve uma solução *IPS* de Código Aberto e gratuita, o *Hogwash Light BR*[1].

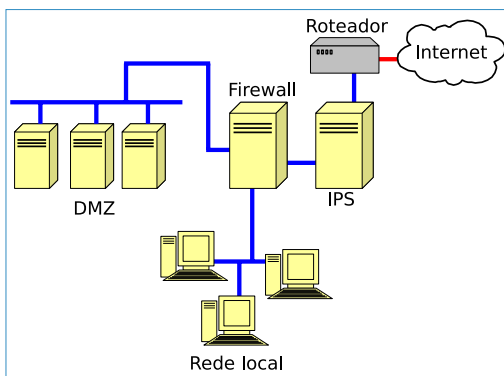


Figura 1 Topologia básica de um ambiente de produção.

## Brasileiro e invisível

O *Hogwash Light BR* é um projeto brasileiro, criado em novembro de 2005, derivado do *Hogwash* (desenvolvido por Jason Larsen em 1996). Esse projeto é destinado à segurança em redes de computadores.

O *HLBR* é capaz de filtrar pacotes diretamente na camada dois do modelo *OSI*[2], o que significa que ele não necessita do endereço *IP* na máquina em que atua. A detecção de tráfego malicioso é baseada em regras simples, e o próprio usuário pode confeccionar novas regras. O *HLBR* é bastante eficiente e versátil, e pode ser usado até mesmo como *bridge* para *honeypots* e *honeynets*. Como não usa a pilha *TCP/IP* do sistema operacional, ele é “invisível” a outras máquinas na rede e atacantes.

## Laboratório

Antes de implementar o *HLBR* para melhorar a segurança de sua rede, é interessante fazê-lo num laboratório, sob condições controladas, a fim de adquirir maior familiaridade com o sistema. Dessa forma, será possível implementá-lo de acordo com as necessidades da rede em questão.

O cenário usado nos testes descritos neste artigo foi composto por três máquinas: uma para o alvo (plataforma Microsoft), outra para o *IPS* e uma terceira para o agressor remoto:

- ◆ Alvo: Windows® XP com *Service Pack 2*, rodando o *IIS 5.1*;

- ◆ *IPS*: *Debian Sarge* (3.1) com *Hogwash Light BR*;
- ◆ Atacante: Qualquer sistema operacional com um interpretador *Python* instalado.

O sistema *Windows* foi utilizado no laboratório devido à implementação em questão, que pedia o uso do mesmo. Um script de *exploit* desse servidor foi criado para fins de teste. Embora esteja disponível em [3], esse *exploit* somente deve ser usado para fins educacionais ou por profissionais de segurança da informação, e jamais com o intuito de prejudicar qualquer pessoa, seja ela física ou jurídica.

São necessários um cabo de rede *cross-over* e duas placas de rede, pois o *HLBR* trabalha como uma *bridge*, ou seja, faz uma ponte entre o *firewall* e o roteador (figura 1). Entretanto, em nosso cenário de testes (figura 2), ele fará a ponte entre o alvo e o atacante.

## Instalação

A configuração de um servidor web *IIS* vai além do escopo deste artigo, assim como a instalação do *Debian Sarge*. Há diversos bons tutoriais disponíveis gratuitamente dedicados a explicar esses processos. Portanto, vamos prosseguir à instalação e configuração do *HLBR*.

Depois de baixar o *Hogwash Light BR* em [4], o arquivo compactado deve ser expandido na máquina que atuará como *IPS*:

```
# wget http://prdownloads.
sourceforge.net/hlbr/hlbr-
1.0.tar.gz
# tar -xvzf hlbr-1.0.tar.gz
```

Em seguida, no diretório onde o HLBR foi descompactado, o comando `configure` prepara a compilação do pacote.

```
# cd hlbr-1.0
# ./configure
```

Será solicitada escolha do idioma. Para selecionar o português, basta pressionar **[P]** seguido de **[Enter]**; para o inglês, basta pressionar **[Enter]**. A compilação e a instalação são realizadas com os tradicionais comandos `make` e `make install`.

```
# make && make install
```

Próxima etapa envolve a eliminação das interfaces de rede. Isso pode ser obtido através da configuração de um novo kernel, sem suporte a TCP/IP; contudo, é mais prático simplesmente atribuir às duas placas de rede da máquina com o IPS endereços IP pertencentes à rede 127.0.0.0 diferentes de 127.0.0.1, pois este pertence à interface `loopback`:

```
# ifconfig eth0 127.0.0.2
# ifconfig eth1 127.0.0.3
```

## Configuração do HLBR

A configuração do HLBR é realizada no arquivo `hlbr.config`. Somente as entradas `IPList` devem ser alteradas, de forma a atender às necessidades de cada ambiente de testes.

Em nosso laboratório, o IP do servidor web é 192.168.0.80; por isso o arquivo de configuração fica conforme o [exemplo 1](#). Nesse exemplo, as entradas `IPList` diferentes de `www` estão apenas comentadas, pois serão necessárias para a configuração definitiva do IPS em ambiente de produção.

## Testes

Finalizada a configuração, vamos aos testes. Primeiro, vamos conferir se o software está corretamente instalado e configurado. As regras de filtragem ficam no diretório `/etc/hlbr/rules/`. Há também um arquivo vazio em `/etc/hlbr/empty.rules`, que pode ser usado para o teste inicial no [exemplo 2](#).

Caso a saída do teste tenha sido semelhante à do [exemplo 2](#), podemos prosseguir ao teste de conectividade, disparando um `ping` da máquina agressora contra o servidor web:

```
at@lab:~$ sudo ping 192.168.0.80
```

Um ping bem-sucedido significa que, no quesito conectividade, as coisas estão bem.

Esperamos que um ataque contra o servidor web tenha sucesso neste momento, já que não há regra de filtragem ativa no IPS. No laboratório, foi disparado o script `iis5hack.py` contra o servidor de testes, que evidentemente estava corretamente iniciado e no ar.

Ao ser atacado, o servidor web exibiu um alerta informando que o software IIS, alvo do teste de agressão, havia sido fechado. Isso confirmou que o IPS de fato estava invisível tanto para a máquina atacante quanto para o alvo do ataque. Sem regras carregadas, o IPS simplesmente encaminhou à máquina-alvo todos os pacotes enviados pelo agressor.

### Exemplo 1: hlbr.conf

```
01 <IPList www>
02 192.168.0.80
03 </list>
04
05 #<IPList dns>
...
09 #<IPList email>
...
13 #<IPList firewall>
...
17 #<IPList network>
...
21 #<IPList others>
...
25 #<IPList servers>
...
```

## Exemplo 2: Regras vazias do HLBR

```
# hlbr -c /etc/hlbr.config -r empty.rules &
Rules file is empty.rule
Loaded 0 rules
Tree is empty
Tree is empty
Tree is empty
Tree is empty
Tree is empty
Tree is empty
eth0: Promiscuous mode enabled.
device eth0 entered promiscuous mode
eth1: Promiscuous mode enabled.
device eth1 entered promiscuous mode
```

## Regras

O passo seguinte envolve a criação de regras no HLBR para evitar esse tipo de ataque contra a máquina-alvo. O arquivo `README.pt_BR` incluído na distribuição do HLBR é um ótimo manual para se aprender a instalar, configurar e até mesmo confeccionar suas próprias regras. Em [\[5\]](#) há também um manual online, com mais instruções para configuração de regras.

O arquivo `/etc/hlbr/rules/iisattacks.rules`, que abrigará as regras específicas contra o tipo de ataque que estamos tentando evitar, deve ficar conforme o [exemplo 3](#).

A [linha 1](#) mostra que a escrita de uma regra é muito simples, entre tags `<rule>` e `</rule>`.

Na [linha 2](#), `ip dst()` determina o IP do alvo – `www` foi definido com o IP do servidor alvo no arquivo `hlbr.config`, no [exemplo 1](#). Da mesma forma, `tcp dst()` ([linha 3](#)) define a porta TCP a ser atacada (80, no caso).

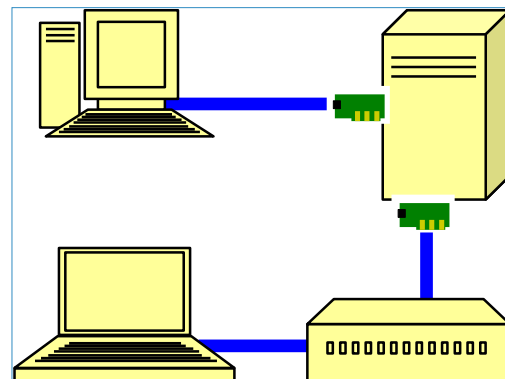


Figura 2 Laboratório simplificado.

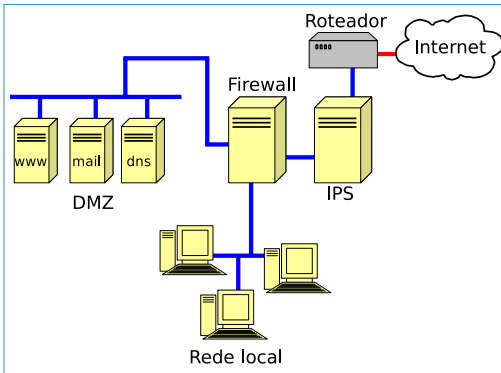


Figura 3 HLBR em ambiente de produção.

Um dos recursos mais poderosos do HLBR é o uso de expressões regulares para impedir ataques, através da escrita de regras mais inteligentes. A **linha 4** define a expressão a ser filtrada, englobando, de uma só vez, múltiplas requisições indicativas do ataque que se deseja impedir.

O uso de **message (linha 5)** permite que o administrador do IPS seja alertado com uma descrição do ataque, e **action (linha 6)** especifica a ação a ser executada. **action1**, como definida em **hlbr.config**, registra o ataque nos arquivos **/var/log/hlbr/hlbr.log** e **/var/log/hlbr/hlbr.dump** e em seguida descarta a tentativa de ataque.

## Em vigor

Vamos parar o daemon do Hogwash Light BR para carregar nossas novas regras:

```
# killall hlbr
# hlbr -c /etc/hlbr/hlbr.config -r
➔rules/iisattacks.rules &
```

Para testar as novas regras, reiniciamos o servidor web na máquina-alvo e disparamos novamente o mesmo ataque contra ela. Acompanhando o arquivo de log do HLBR é possível verificar que o ataque está sendo ativamente bloqueado pelo IPS.

## Totalmente seguros?

Na verdade, não existe software totalmente seguro, e sim aqueles cujas vulnerabilidades ainda não foram

### Exemplo 3: iisattacks.rules

```
01 <rule>
02 ip dst(www)
03 tcp dst(80)
04 tcp regex(^http://.+(_vti_bin|_sharepoint)+(/.dll)+.[0-9]$/)
05 message=(iisattacks) Ataque ao IIS 5.1
06 action=action1
07 </rule>
```

### Exemplo 4: apacheattacks.rules

```
01 <rule>
02 ip dst(192.168.0.100)
03 tcp dst(80)
04 tcp nocase([90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90])
05 message=(apacheattacks) Ataque ao Apache 1.3.27
06 action=action1
07 </rule>
```

### Exemplo 5: hlbr.rules

```
01 <include rules/awstats.rules>
02 <include rules/bufferoverflow1.rules>
...
13 <include rules/iisattacks.rules> # Arquivo de regras do exemplo 5
14 <include rules/apacheattacks.rules> # Arquivo de regras do exemplo 6
```

descobertas. Por exemplo, o próprio *Apache* (versão 1.3.27) possui uma falha de estouro de *buffer* que é disparada se o servidor receber um pacote com o seguinte conteúdo hexadecimal:

```
|90 90 90 90 90 90 90 90 90 90 90
➔90 90 90 90 90|
```

Com isso, poderíamos escrever uma regra no HLBR contra esse ataque, colocando-a, por exemplo, no arquivo **/etc/hlbr/rules/apacheattacks.rules**, conforme o **exemplo 4**.

Feito isso, esse ataque específico falharia ao atravessar o IPS.

## Ambiente de produção

Após o administrador se familiarizar com a configuração e criação de regras do HLBR, ele já pode instalá-lo num ambiente de produção. Nosso cenário de produção é ilustrado pela **figura 3**, e inclui um firewall, três servidores numa DMZ e uma rede local, além, é claro, do IPS, localizado entre o firewall e o roteador que conecta a rede à Internet.

No cenário acima, usaremos os seguintes IPs (eles não são reais):

- ♦ firewall: 200.200.200.1
- ♦ dns: 200.200.200.2
- ♦ mail: 200.200.200.3
- ♦ www: 200.200.200.4

O primeiro passo para a implementação é customizar o arquivo **hlbr.config**. Nele, adicionaremos os IPs de nossos servidores públicos (**www**, **mail** e **dns**) que precisam ser protegidos pelo HLBR. Para isso, deve-se procurar a seção **IP Lists** e fazer, a seguir, as seguintes alterações:

```
<IPList www>
200.200.200.4
</list>
<IPList dns>
200.200.200.2
</list>
<IPList email>
200.200.200.3
</list>
<IPList firewall>
200.200.200.1
</list>
<IPList servers>
www
dns
email
```

```
firewall
18 </list>
```

O arquivo `hlbr.rules` vem com 86 regras prontas para usar, bastando somente iniciar o `daemon` do HLBR com os arquivos de regras contidos nele:

```
# hlbr -c hlbr.config -r hlbr.
rules &
```

ou

```
# /etc/init.d/hlbr start
```

Devem aparecer algumas linhas na tela informando que foram carregadas todas as 86 regras e dizendo que as interfaces de rede entraram no modo promíscuo, necessário à operação do HLBR. Depois disso, o HLBR já estará rodando e aumentando a segurança da rede.

## Organização

Qualquer ferramenta de segurança deve permitir a organização de seus arquivos de configuração – afinal, desorganização facilita a insegurança. O HLBR permite a criação de regras de filtragem em arquivos diversos, e a referência aos mesmos a partir de um arquivo central, `/etc/hlbr/hlbr.rules` (exemplo 5).

Após qualquer alteração às regras ou à configuração do HLBR, é necessário reiniciar o daemon.

## Conclusões

Todo software, por melhor que seja, tem falhas. Garantir a segurança de servidores passa pela inativação de suas vulnerabilidades. O HLBR é um grande aliado nessa tarefa, pois pode impedir que ataques conhecidos sejam desferidos contra máquinas da rede, evitando assim maiores conseqüências. ■

## Mais Informações

- [1] Hogwash Light BR: <http://hlbr.sourceforge.net>
- [2] Modelo OSI: [http://pt.wikipedia.org/wiki/Modelo\\_OSI](http://pt.wikipedia.org/wiki/Modelo_OSI)
- [3] Exploit para servidor IIS 5.1: <http://www.rogerioferreira.objectis.net/downloads/iis5hack.py>
- [4] Download do HLBR: <http://sourceforge.net/projects/hlbr>
- [5] Manual online do HLBR: <http://hogwash.sourceforge.net/docs/index.htm>

## O autor

Precursor da implantação de Software Livre no Tribunal de Contas do Estado do Amazonas na área de Segurança de Redes, **Rogério Ferreira** (rogerio@cumorahsystems.com) trabalha atualmente na Cumorah Systems, empresa especializada em Segurança da Informação.



# PeP link

## Acesso confiável à Internet com alto desempenho e baixo custo.

A linha de roteadores profissionais **PePLink** leva para sua empresa recursos avançados de rede com um custo acessível.

O **PePLink** permite conexão simultânea com até sete links de Internet, balanceamento de carga entre os links, firewall e servidor DNS.

Estes recursos permitem uma conexão com a Internet estável e contínua mesmo quando ocorrem problemas com links de Internet.

A interface de configuração e monitoramento do **PePLink** é fácil de usar e intuitiva. Não é necessário ter conhecimentos avançados em rede para utilizar os recursos avançados do **PePLink**.

O **PePLink** é a solução para manter e controlar links redundantes de forma descomplicada e barata.

