HLBR - O emprego de uma bridge como IPS para a segurança em redes de computadores

André Bertelli Araújo[1], João Eriberto Mota Filho[2]

[1]Serviço Federal de Processamento de Dados (SERPRO)
SGAN 601 - Módulo G
70830-900 - Brasília - DF

[2]Comando do Exército Brasileiro
Quartel General do Exército - Bloco A - 3º Piso
70630-901 - Brasília - DF

andre@bertelli.name, eriberto@eriberto.pro.br

Abstract. *HLBR (Hogwash Light BR) is a free/libre IPS, developed in Brazil, based in Jason Larsen's Hogwash, which captures data directly from layer 2 of the OSI model (link layer). It works as a bridge, being able to detect and handle malicious traffic. HLBR is invisible to attackers, since it doesn't change the packets' headers. The project initial goal was to refine Hogwash's code, in order to make it more functional. Some planned enhancements were already implemented, including the use of regular expressions. This document will describe HLBR's main features and uses, as well as all the research work done to enhance it and make it an efficient IPS.*

Resumo. *O HLBR (Hogwash Light BR) é um IPS livre, desenvolvido no Brasil, baseado no Hogwash de Jason Larsen, que colhe dados diretamente na camada 2 do modelo OSI (camada de enlace). Atua como uma bridge, sendo capaz de detectar e tratar tráfego malicioso. O HLBR é invisível aos olhos de um atacante, uma vez que não altera o cabeçalho dos pacotes. O objetivo inicial do projeto foi refinar o código do Hogwash, a fim de torná-lo mais funcional. Uma série de melhorias previstas já foram implementadas, inclusive o uso de expressões regulares. Este documento descreverá as principais características e aplicações do HLBR, bem como toda a pesquisa realizada para aperfeiçoá-lo e torná-lo um IPS eficiente.*

1. Introdução

As redes de computadores sofrem constantes ataques que buscam a negação do serviço, a intrusão em servidores e o roubo de informações que, segundo Chapman, Cooper e Zwich (2000), são as três categorias básicas de ataques remotos. É grande a quantidade de pessoas sem qualquer conhecimento relativo à área de segurança que tenta realizar ações maliciosas contra as redes. Mesmo sem o esclarecimento necessário, essas pessoas representam um perigo em potencial, pois buscam utilizar, contra as redes, exploits e procedimentos malignos, facilmente encontrados na Internet, podendo causar danos letais. Há também a possibilidade de o atacante em questão ser experiente.

Da necessidade de proteger as redes de computadores nasceram os sistemas de firewall. Um sistema de firewall é composto por vários elementos que atuam de forma

diferente mas com o objetivo comum de prover segurança. Os elementos mais comuns em sistemas de firewall são os filtros de pacotes, os filtros de estados, os proxies e os IPS [Nakamura e Geus 2003].

O IDS (Intrusion Detection System) é um elemento que detecta e registra em log o tráfego malicioso, sem tratá-lo. Há alguns anos, surgiu o conceito de IDS reativo que, ao detectar o tráfego malicioso, enviava um sinal reset para ambos os lados da conexão (atacante e atacado). No entanto, quando o sinal reset era enviado, pelo menos um pacote malicioso já haveria chegado ao seu destino. Assim sendo, essa modalidade de detecção combinada com reação foi rejeitada rapidamente pela comunidade de segurança em redes. A solução foi a criação do IPS (Intrusion Prevention System), um sistema in-line na topologia capaz de detectar e tratar anomalias de tráfego com um nível maior de segurança e eficiência [Nakamura e Geus 2003].

2. O Projeto HLBR

O Hogwash, desenvolvido por Jason Larsen, surgiu como um projeto universitário que utilizava o mecanismo de detecção do IDS Snort, atuando na camada 2 do modelo OSI. Com o tempo, várias modificações ocorreram, dando origem a um formato próprio de regras e detecção. O grande problema é que, apesar de ainda existir o Projeto Hogwash, o seu desenvolvimento foi abandonado há cerca de 2 anos. Com isso, alguns bugs não foram sanados e novas implementações não foram feitas.

O HLBR é um software livre que utiliza o Hogwash como base. O projeto, fundado em novembro de 2005, inicialmente, retirou bugs do Hogwash e algumas funcionalidades julgadas não tão importantes para o novo projeto. A versão 0.1 final, lançada em 26 de dezembro de 2005, implementou, dentre outras novidades, um sistema de instalação automatizado e um documento do tipo README com todas as informações básicas necessárias para a operação do IPS. A versão 0.2, lançada em 09 de fevereiro de 2006, fez diversas correções e modificações para um melhor funcionamento do sistema. A versão 1.0, lançada em 05 de março de 2006, implementou o uso de expressões regulares nas regras, dando ao HLBR mais flexibilidade e poder de detecção de anomalias no tráfego. A versão 2.0, a ser lançada até o final de 2006, implementará o uso de inteligência artificial sobre o tráfego que já tenha sido analisado pelas regras convencionais e que tenha sido considerado normal.

O HLBR está sendo desenvolvido para o Sistema Operacional GNU/Linux. No entanto, poderá ser compilado em BSDs e adaptado a outros sabores de Unix.

3. O posicionamento do HLBR

Por ser um IPS, o HLBR deve ser posicionado in-line na topologia. Por ser uma bridge, o mesmo irá repassar os pacotes sem a necessidade de roteamento de rede. A Figura 1 mostra o posicionamento do HLBR em uma rede simples.

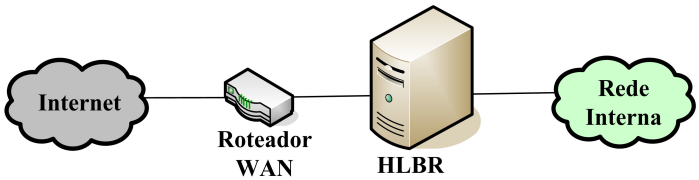

Figura 1: Posicionamento do HLBR.

Dentro de um sistema de firewall corporativo, o HLBR poderá ser colocado em várias posições diferentes. Poderá haver, por exemplo, uma proteção dos servidores contra ataques internos e externos. A Figura 2 exemplifica essa situação, mostrando o HLBR dentro de um sistema de firewall, delimitado por uma linha tracejada.

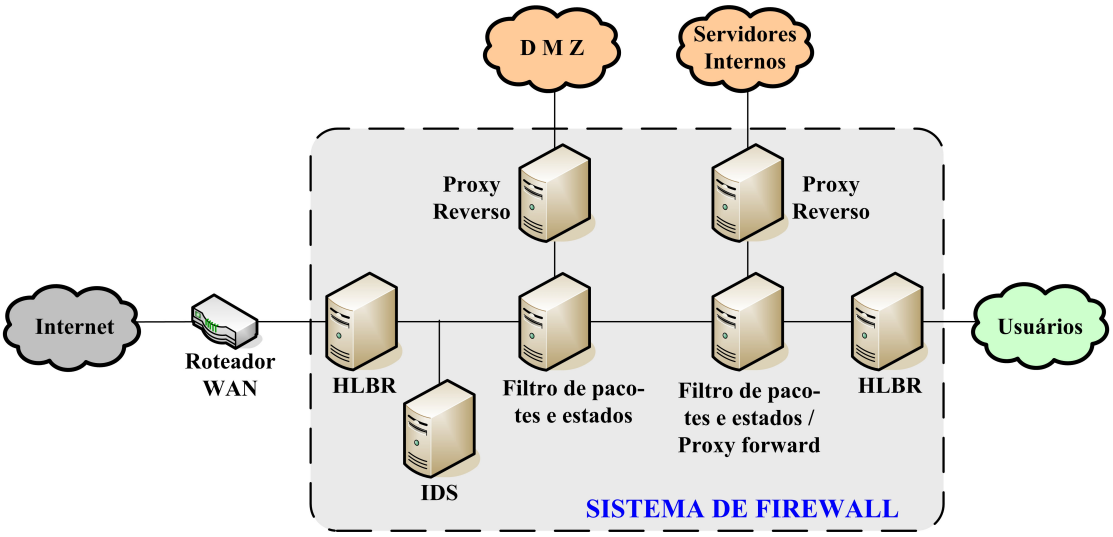

Figura 2: HLBR, dentro de um sistema de firewall, protegendo contra ataques externos e internos.

Na Figura 2 o HLBR aparece em posições estratégicas. Nessas posições ele poderá ser utilizado não só para bloquear pacotes maliciosos mas também para realizar log de tráfego suspeito para futura análise. É importante citar que há um IDS posicionado entre a Internet e a rede. Os logs desse IDS, com certeza, serão úteis no momento da confecção de novas regras para o IPS.

Outra possibilidade do HLBR é a de servir como ponte para uma honeynet, cujo objetivo é capturar dados provenientes de ataques para estudos [Northcutt 2002] que também resultarão na implementação de novas regras. Isso está evidenciado na Figura 3. Essa funcionalidade foi herdada do Hogwash e encontra-se em fase final de reajuste.

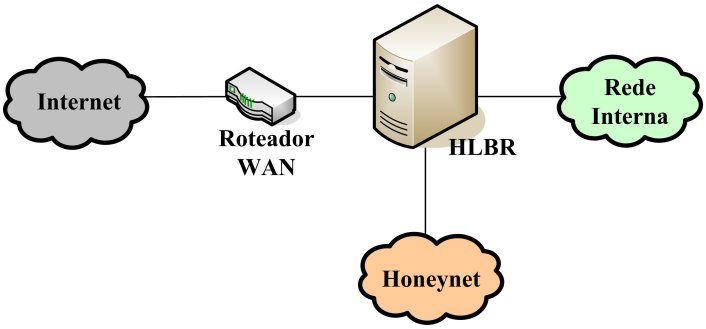

Figura 3: HLBR como ponte para uma honeynet.

O uso de uma honeynet com o HLBR é interessante pois, uma vez que o mesmo faz toda a passagem de tráfego pela camada de enlace, é possível utilizar em tal honey os mesmos endereços IP existentes nos servidores de rede. Com isso, um atacante sempre receberá respostas oriundas dos endereços IP atacados, independentemente de ter sido desviado para a honeynet ou não.

4. Os adaptadores de rede no HLBR

O HLBR atua como uma bridge e, portanto, não necessita de endereço IP nos adaptadores de rede. No entanto, para que isso ocorra, deverá haver uma recompilação do Kernel Linux a fim de remover toda a parte TCP/IP do mesmo, disponível atualmente no caminho Networking > Networking options > TCP/IP networking. Isso será necessário porque o TCP/IP irá conflitar com o fato dos adaptadores de rede estarem ativos sem endereço IP. Assim, o HLBR poderá ter problemas no seu funcionamento, como retardos no tráfego etc.

Há uma segunda opção que é a atribuição de endereços IP da rede 127.0.0.0/8 aos adaptadores de rede. Essa opção é segura e tem a vantagem de permitir a análise do tráfego na camada 3 do modelo OSI (camada de rede) com ferramentas de rede como IPTraf e TCPDUMP. Até mesmo um IDS, como o Snort, poderá colher dados para análise, se instalado na mesma máquina do HLBR. A vantagem será o fato do IDS estar in-line na topologia. Os seus logs poderão ser utilizados para embasar a confecção de novas regras. No entanto, isso exigirá um hardware potente, uma vez que o processador estará ocupado com uma dupla análise de cada pacote.

5. O funcionamento do HLBR

O HLBR trabalha com dois tipos de logs: um deles registra a anomalia detectada e o outro um dump do tráfego. A seguir, dois exemplos do primeiro tipo de log (alguns octetos foram mascarados):

```
00831 22/02/2006 13:23:00 x.51.162.252:3324->y.z.48.244:80 (http-2-re) open
      proxy search
00832 22/02/2006 13:25:55 x.175.55.194:3063->y.z.48.242:80 (webattacks-2-
      re) directory change attempt (unicode,asc,plain)
```

A seguir, o log do tipo dump referente aos dois casos anteriores:

```
13:23:00.661777 IP x.51.162.252.3324 > y.z.48.244.www: P
      1448357854:1448357917(63) ack 1691128774 win 17520
        0x0000:  4500 0067 3a0e 4000 6b06 02c6 d233 a2fc  E..g:.@.k....3..
        0x0010:  c8fc 9490 0cfc 0050 5654 2fde 64cc 93c6  .......PVT/.d...
        0x0020:  5018 4470 30d1 0000 434f 4e4e 4543 5420  P.Dp0...CONNECT.
        0x0030:  3231 302e 3738 2e31 3438 2e31 3632 3a32  xxx.78.148.162:2
        0x0040:  3520 4854 5450 2f31 2e31 0d0a 486f 7374  5.HTTP/1.1..Host
        0x0050:  3a20 3231 302e 3738 2e31 3438 2e31 3632  :.xxx.78.148.162
        0x0060:  3a32 350d 0a0d 0a                        :25....
13:25:55.965631 IP x.175.55.194.3063 > y.z.48.242.80: P 0:339(339) ack 1
      win 1452 <nop,nop,timestamp 1367990529 2076841557>
        0x0000:  4500 0187 a264 4000 3a06 770f c8af ffc2  E....d@.:.w.....
        0x0010:  c8fc 948e 0bf7 0050 6d60 2427 8b80 03ef  .......Pm`$'....
        0x0020:  8018 05ac 3f67 0000 0101 080a 5189 e101  ....?g......Q...
        0x0030:  7bca 1655 4745 5420 2f2e 2e2f 2e2e 2f65  {..UGET./../../e
        0x0040:  7463 2f20 4854 5450 2f31 2e30 0d0a 486f  tc/.HTTP/1.0..Ho
        0x0050:  7374 3a20 7777 772e 6761 6263 6d74 2e65  st:.www.xxxxxx.y
        0x0060:  622e 6d69 6c2e 6272 0d0a 4163 6365 7074  y.zzz.br..Accept
        0x0070:  3a20 7465 7874 2f68 746d 6c2c 2074 6578  :.text/html,.tex
        0x0080:  742f 706c 6169 6e2c 2061 7070 6c69 6361  t/plain,.applica
        0x0090:  7469 6f6e 2f78 2d74 726f 6666 2d6d 616e  tion/x-troff-man
        0x00a0:  2c20 6170 706c 6963 6174 696f 6e2f 782d  ,.application/x-
        0x00b0:  7461 722c 2061 7070 6c69 6361 7469 6f6e  tar,.application
        0x00c0:  2f78 2d67 7461 722c 2074 6578 742f 2a2c  /x-gtar,.text/*,
        0x00d0:  2061 7070 6c69 6361 7469 6f6e 2f78 2d64  .application/x-d
        0x00e0:  6562 6961 6e2d 7061 636b 6167 652c 2061  ebian-package,.a
        0x00f0:  7564 696f 2f62 6173 6963 2c20 2a2f 2a3b  udio/basic,.*/*;
        0x0100:  713d 302e 3031 0d0a 4163 6365 7074 2d45  q=0.01..Accept-E
        0x0110:  6e63 6f64 696e 673a 2067 7a69 702c 2063  ncoding:.gzip,.c
        0x0120:  6f6d 7072 6573 730d 0a41 6363 6570 742d  ompress..Accept-
```

```
0x0130:  4c61 6e67 7561 6765 3a20 656e 0d0a 5573  Language:.en..Us
0x0140:  6572 2d41 6765 6e74 3a20 4c79 6e78 2f32  er-Agent:.Lynx/2
0x0150:  2e38 2e35 7265 6c2e 3120 6c69 6277 7777  .8.5rel.1.libwww
0x0160:  2d46 4d2f 322e 3134 2053 534c 2d4d 4d2f  -FM/2.14.SSL-MM/
0x0170:  312e 342e 3120 474e 5554 4c53 2f31 2e30  1.4.1.GNUTLS/1.0
0x0180:  2e31 360d 0a0d 0a                        .16....
```

No primeiro caso houve uma tentativa de ataque sobre um servidor web, buscando utilizá-lo como open proxy. Isso ficou caracterizado com o trecho de tráfego "`CONNECT xxx.78.148.162:25 HTTP/1.1`". No caso seguinte houve uma tentativa de mudança de diretórios também em um servidor web. O tráfego característico foi "`GET /../../etc/ HTTP/1.0`". Ressalta-se que o HLBR pode analisar qualquer tráfego TCP, UDP e ICMP, podendo filtrar pacotes destinados a servidores de e-mail, DNS, chat etc.

No HLBR existem as chamadas ações. Cada ação tem um nome é caracterizada por um conjunto de procedimentos que serão adotados caso alguma regra detecte a passagem de um pacote malicioso. As ações mais comuns são o registro em log e o descarte do pacote. A seguir, um exemplo de uma ação, denominada `action1`, que faz os logs de registro e dump, além do descarte de pacotes maliciosos:

```
<action action1>
response=alert file(/var/log/hlbr/hlbr.log)
response=dump packet(/var/log/hlbr/hlbr.dump)
response=drop
</action>
```

O HLBR, atualmente, funciona com base em regras de detecção. Essas regras poderão ser confeccionadas pelo próprio usuário, se assim o desejar. No entanto, o IPS já é fornecido com dezenas de regras. A seguir, as duas regras que detectaram o tráfego anômalo mostrado nos logs, ambas utilizando expressões regulares:

```
<rule>
ip dst(www)
tcp dst(80)
tcp regex(^CONNECT )
message=(http-2-re) open proxy search
action=action1
</rule>
<rule>

ip dst(www)
tcp dst(80)
tcp regex((GET|POST|PUT) [^\n]+(/|%252f|%2f|\\|%255c|%5c)(\.|%252e|
      %2e)+(/|%252f|%2f|\\|%255c|%5c)+)
message=(webattacks-2-re) directory change attempt (unicode,asc,plain)
action=action1
</rule>
```

As regras anteriores estabeleceram bloqueios baseados em expressões regulares. As duas referiram-se ao tráfego TCP destinado à porta 80 dos servidores web, representados por www. A tag www foi referenciada dentro do arquivo de configuração principal do HLBR e a ela foi atribuído o endereço IP de três servidores web. A primeira regra enquadrou o tráfego HTTP que contivesse a expressão "CONNECT " no início do pacote. A segunda regra enquadrou o tráfego HTTP que contivesse GET, POST ou PUT, seguido de seqüências de barras, barras invertidas e pontos ou os seus respectivos códigos unicode, com o intuito de evitar a mudança forçada de diretório dentro de um servidor web. Com isso, seqüências como "`/../../../`", "`/./../.../`", "`/./..//...///`", "`..\..\../../../`" e "`%252f..%252f..`" seriam bloqueadas.

Como mais um exemplo, a linha "`tcp regex(filename="[^\n]+\.exe")`" evitaria que arquivos com extensão `.exe` chegassem ao servidor de e-mail.

6. Conclusão

O HLBR é um IPS que atua na camada 2 do modelo OSI. Por ser uma bridge, não modifica o cabeçalho dos pacotes que por ele passam. Isso faz com que o mesmo seja invisível aos olhos de atacantes. Outro aspecto que o torna transparente é o fato dele descartar o pacote malicioso, sem resetar ou finalizar a conexão, deixando a origem do ataque sem saber o que está ocorrendo e induzindo-a a pensar que algo está errado com o exploit ou outro artifício utilizado. Esse tráfego também poderá ser desviado para uma honeynet, de forma a iludir o atacante e registrar os seus atos.

O HLBR tem o poder de analisar o conteúdo dos pacotes que por ele passam. Por ser um IPS, o HLBR deve ser posicionado in-line na topologia. Logicamente, haverá uma limitação no que tange ao tráfego criptografado. Nesse caso, um HIDS deverá ser utilizado na máquina destino para tentar detectar anomalias [Nakamura e Geus 2003].

Quanto ao projeto, constantes estudos estão ocorrendo para gerar cada vez mais assinaturas de ataques ou de tráfego não permitido. Dois integrantes da equipe, ambos mestres com dissertação voltada para a área de detecção de intrusões com o uso de inteligência artificial, estão dedicando-se integralmente à incorporação desse tipo de recurso no HLBR. Cabe ainda ressaltar que várias partes do projeto já estão traduzidas para treze idiomas.

Referências

Chapman, D. Brent; Cooper, Simon; Zwicky, Elizabeth D. (2000) "Building Internet Firewalls", O´Reilly & Associates Inc., 2nd edition, page 11.

HLBR. (2005) "Hogwash Light BR", http://hlbr.sourceforge.net, acesso em 22 fev. 2006.

IPTraf. (2005) "IPTraf IP Networking Monitoring Software", http://iptraf.seul.org, acesso em 22 fev. 2006.

Jargas, Aurélio M. (2001) "Expressões Regulares - Guia de Consulta Rápida", Novatec, 96p.

Larsen, Jason. (2003) "Welcome to Hogwash", http://hogwash.sourceforge.net, acesso em 22 fev. 2006.

Nakamura, Emílio Tissato; Geus, Paulo Lício de. (2003) "Segurança de Redes em Ambientes Cooperativos", Futura, 2ª edição, páginas 256-257.

Northcutt, Stephen; Novak, Judy. (2002) "Network Intrusion Detection", New Riders Publishing, 3rd edition, page 264.

Stevens, Richard. (1994) "TCP/IP Illustrated, Volume 1: The Protocols", Addison-Wesley, 576p.

TCPDUMP. (2005) "TCPDUMP/LIBPCAP", http://www.tcpdump.org, acesso em 22 fev. 2006.