interNexo
Consultoria e Serviços de Acesso à Internet

fwbuilder – Firewall Builder

Uniformizando a configuração de firewalls

Ethy H. Brito
dez/2006

interNexo
Consultoria e Serviços de Acesso à Internet

- FWB – o que é e como encontrar
- Instalação
- Modelo abstrato
- Vantagens
- Interface Gráfica
- Exemplos de Scripts
- Conclusões

interNexo
Consultoria e Serviços de Acesso à Internet

FWB – O que é e onde encontrar

- Código fonte aberto - Open Source (exceto Cisco PIX)
- Gerador de scripts a partir de interface gráfica (object-oriented)
- Usa modelo abstrato próprio
- Regras para:
  Iptables, (Linux) ,
  PIX (Cisco),
  IPFW, PF (*BSD)
- Disponível em: http://www.fwbuilder.org/
- Desenvolvido por: Vadim Kurland (vadim@fwbuilder.org)
- Versão atual: 2.1.8
- Lista de discussão: fwbuilder-discussion@lists.sourceforge.net

interNexo
Consultoria e Serviços de Acesso à Internet

Instalação

- Pacotes
  Windows, MacOSX, Fedora, FreeBSD, Mandrake, RedHat e Suse

- Fontes
  Biblioteca de “compiladores”
  Interface Gráfica

- Dependências
  libxml2 v2.4.10 ou + nova
  libxslt v1.0.7 ou +nova
  ucd-snmp ou net-snmp (opcional)
  openssl (sempre usar a versão mais nova)
  QT 3.1.x, 3.2.x, 3.3.x

- Contribuições: scripts de ajuda desenvolvidos por terceiros

- Instalação segue o processo CMMi
  (configure; make; make install)

interNexo Consultoria e Serviços de Acesso à Internet

Modelo Abstrato

- Superconjunto das propriedades das plataformas suportadas
- Objetos reaproveitáveis: Firewall, Objects (hosts, endereços, e seus agrupamentos), Services e Time
- Regras podem ser “atribuídas” a uma interface ou não (automaticamente atreladas a uma cadeia - chain)
- Seleção da cadeia baseada em *origem-destino-serviço-ação*
- Regras *stateful* emuladas em plataforma não conformes
- Regras com múltiplos “objects” (mesmo com “negação”);
- Simplificações e otimizações são decorrentes do modelo
- Não evita a existência de regras impossíveis em algumas plataformas (falta de suporte básico na plataforma alvo)

Fragmento do Arquivo de políticas

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE FWObjectDatabase SYSTEM "fwbuilder.dtd">
<FWObjectDatabase xmlns="http://www.fwbuilder.org/1.0/" version="2.0.12"
lastModified="1165438898" id="root">

...
   <PolicyRule action="Accept" disabled="False" id="id415C388D" log="False"
         position="1">
      <Src neg="False">
         <ObjectRef ref="sysid0"/>
      </Src>
      <Dst neg="False">
         <ObjectRef ref="id415C1848"/>
      </Dst>
      <Srv neg="False">
         <ServiceRef ref="tcp-HTTP"/>
         <ServiceRef ref="id3B4FED69"/>
         <ServiceRef ref="udp-DNS"/>
      </Srv>
      <When neg="False">
         <IntervalRef ref="sysid2"/>
      </When>
      <PolicyRuleOptions>
         <Option name="color"></Option>
      </PolicyRuleOptions>
   </PolicyRule>
...
```

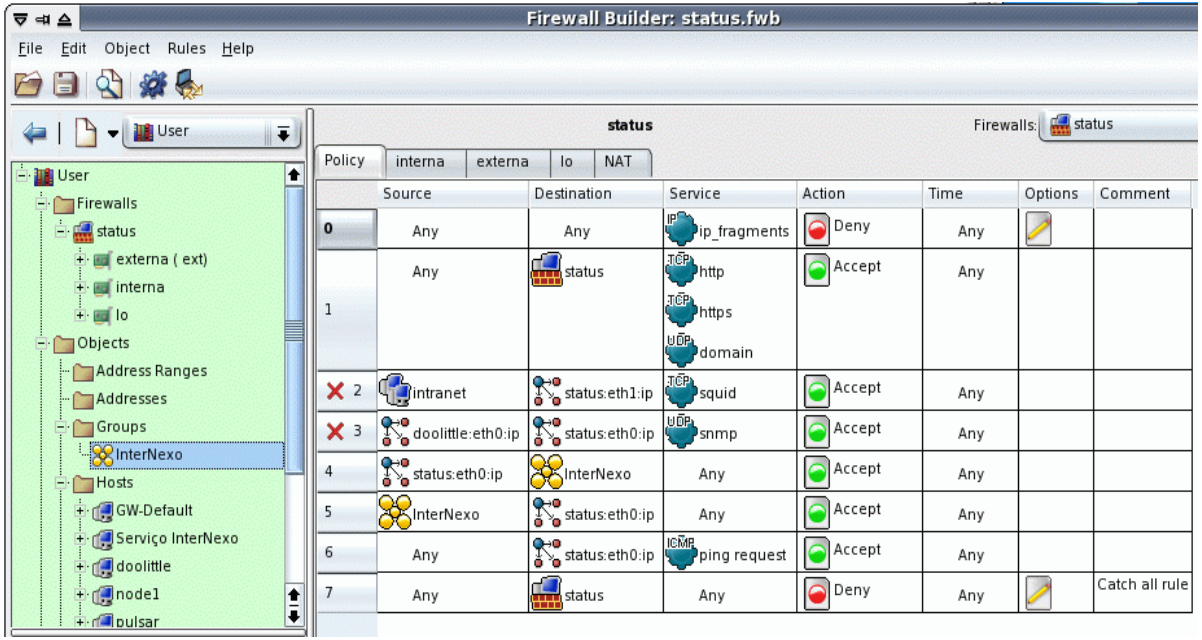
interNexo Consultoria e Serviços de Acesso à Internet

Vantagens

- Migração entre plataformas com baixo impacto;
- Ganho em produtividade – menos detalhes para cuidar;
- Conjunto Quase-mínimo de regras - otimização;
- Erros mais comuns são apontados na compilação antes da instalação das regras;
- Conjunto de objetos de uma mesma companhia mantidos juntos
- Detecção de “sobreamento” de regras (rules shadowing)
- Mudanças em um objeto são imediatamente refletidas em todas as regras
- Lei de Lavoisier adaptada à informática

interNexo
Consultoria e Serviços de Acesso à Internet

Interface Gráfica



interNexo
Consultoria e Serviços de Acesso à Internet

Exemplo de regras geradas para IPTABLES

```
# Rule 0 (global)
#
echo "Rule 0 (global)"
#
$IPTABLES -N RULE_0
$IPTABLES -A OUTPUT -p all  -f   -j RULE_0
$IPTABLES -A INPUT -p all  -f   -j RULE_0
$IPTABLES -A FORWARD -p all  -f   -j RULE_0
$IPTABLES -A RULE_0  -j LOG  --log-level info --log-prefix "RULE 0 -- DENY "
$IPTABLES -A RULE_0  -j DROP
#
# Rule 1 (global)
#
echo "Rule 1 (global)"
#
$IPTABLES -N Cid415C388D.0
$IPTABLES -A OUTPUT  -d 192.168.0.254  -m state --state NEW  -j Cid415C388D.0
$IPTABLES -A OUTPUT  -d 200.231.48.37  -m state --state NEW  -j Cid415C388D.0
$IPTABLES -A Cid415C388D.0 -p tcp -m tcp -m multiport --dports 80,443 -j
ACCEPT
$IPTABLES -A Cid415C388D.0 -p udp -m udp  --dport 53  -j ACCEPT
$IPTABLES -N Cid415C388D.1
$IPTABLES -A INPUT  -d 192.168.0.254  -m state --state NEW  -j Cid415C388D.1
$IPTABLES -A INPUT  -d 200.231.48.37  -m state --state NEW  -j Cid415C388D.1
$IPTABLES -A Cid415C388D.1 -p tcp -m tcp  -m multiport  --dports 80,443  -j
ACCEPT
$IPTABLES -A Cid415C388D.1 -p udp -m udp  --dport 53  -j ACCEPT
```

interNexo
Consultoria e Serviços de Acesso à Internet

Continuação

```
#
# Rule 4 (global)
#
echo "Rule 4 (global)"
#
$IPTABLES -N Cid415C38EC.0
$IPTABLES -A OUTPUT  -s 200.231.48.37  -m state --state NEW  -j Cid415C38EC.0
$IPTABLES -A Cid415C38EC.0  -d 200.231.48.38  -j ACCEPT
$IPTABLES -A Cid415C38EC.0  -d 200.231.48.33  -j ACCEPT
$IPTABLES -A Cid415C38EC.0  -d 200.231.48.43  -j ACCEPT
$IPTABLES -A Cid415C38EC.0  -d 200.231.49.241  -j ACCEPT
```

interNexo Consultoria e Serviços de Acesso à Internet

Regras geradas para PF (*BSD)

```
# Tables: (5)
table <id45774174.2> { 192.168.0.254 , 200.231.48.37 }
table <id45774461.2> { 200.231.48.38 , 200.231.48.33 , 200.231.48.43 ,
  200.231.49.241 }

#
# Rule  0 (global)
#
#
block in   log  quick inet  from any  to any  fragment  label "RULE 0 -- DROP
"
block out  log  quick inet  from any  to any  fragment  label "RULE 0 -- DROP
"
#
# Rule  1 (global)
#
#
pass in   quick inet proto tcp  from any  to <id45774174.2> port { 80, 443 }
flags S/SA keep state  label "RULE 1 -- ACCEPT "
pass in   quick inet proto udp  from any  to <id45774174.2> port 53 keep
state  label "RULE 1 -- ACCEPT "
```

Continuação

```
#
# Rule  4 (global)
#
#
pass in   quick inet  from 200.231.48.37  to <id45774461.2> keep state  label
"RULE 4 -- ACCEPT "
pass out  quick inet  from 200.231.48.37  to <id45774461.2> keep state  label
"RULE 4 -- ACCEPT "
```

Exemplo de regras geradas para IPFW

```
#
# Rule 0 (global)
#
"$IPFW" add 280 set 1 drop    log all  from any  to any   frag      || exit 1

#
# Rule 1 (global)
#
"$IPFW" add 290 set 1 permit tcp  from any  to me 80,443 in  setup keep-state
|| exit 1

"$IPFW" add 300 set 1 permit udp  from any  to me 53 in  keep-state  || exit
1

#
# Rule 4 (global)
#
"$IPFW" add 310 set 1 permit all  from 200.231.48.37  to 200.231.48.38
keep-state  || exit 1

"$IPFW" add 320 set 1 permit all  from 200.231.48.37  to 200.231.48.33
keep-state  || exit 1

"$IPFW" add 330 set 1 permit all  from 200.231.48.37  to 200.231.48.43
keep-state  || exit 1

"$IPFW" add 340 set 1 permit all  from 200.231.48.37  to 200.231.49.241
keep-state  || exit 1
```

Conclusões

FwBuilder é uma ferramenta de fácil manuseio que agiliza o desenvolvimento e administração de filtros de pacotes sobremaneira.

O uso dado para ela não se restringe a pequenas instalações. Temos notícias de instalações compostas de algumas centenas de regras onde a ferramenta tem se saído muito bem.

Existem outros esforços para administrar um filtro e suas tarefas de projeto de políticas (veja referências abaixo). Cada uma delas com alguma pequena deficiência que podemos usar para classificá-las um pouco abaixo do FWB quanto ao quesito completude.

Referências

As referências aqui apresentadas foram todas informadas pelo autor da ferramenta, sr. Vadim Kurland.

- cp2fwbuilder, data file converter from Checkpoint Firewall 1 to FwBuilder, http://cp2fwbuilder.sourceforge.net/
- fwboptimizer: Optimizer for fwbuilder http://gd.tuwien.ac.at/opsys/linux/sf/subcat/tim/fwboptimizer/
- High Level Firewall Language, http://www.hlfl.org/
- FireHOL, the iptables stateful packet filtering firewall builder. http://firehol.sourceforge.net/
- Ipfilter ruleset editor and remote management tool, http://inc2.com/isba/
- Solsoft NP, a suite of policy management solutions for network security, http://www.solsoft.com/