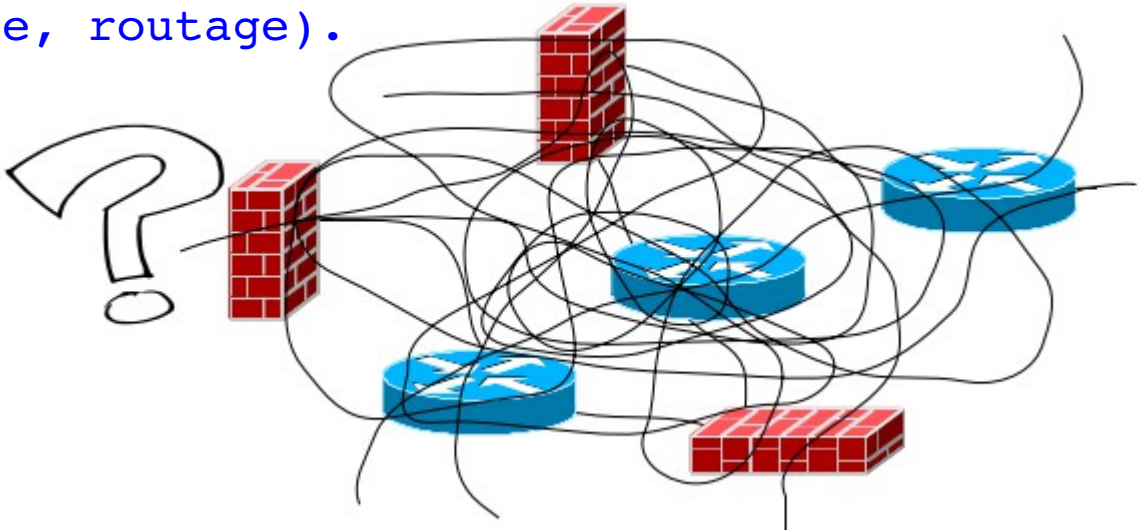
Lsfw : outil de tests de règles de pare-feu distribués sur un réseau

Patrick Lamaizière
Centre de Ressources Informatiques
Université de Rennes 1

lsfw

- Filtrage assuré par des pare-feu distribués sur l'ensemble du réseau ;
- Cumul des règles de filtrage ;
- Difficulté de vérifier la politique de sécurité de bout-en-bout.

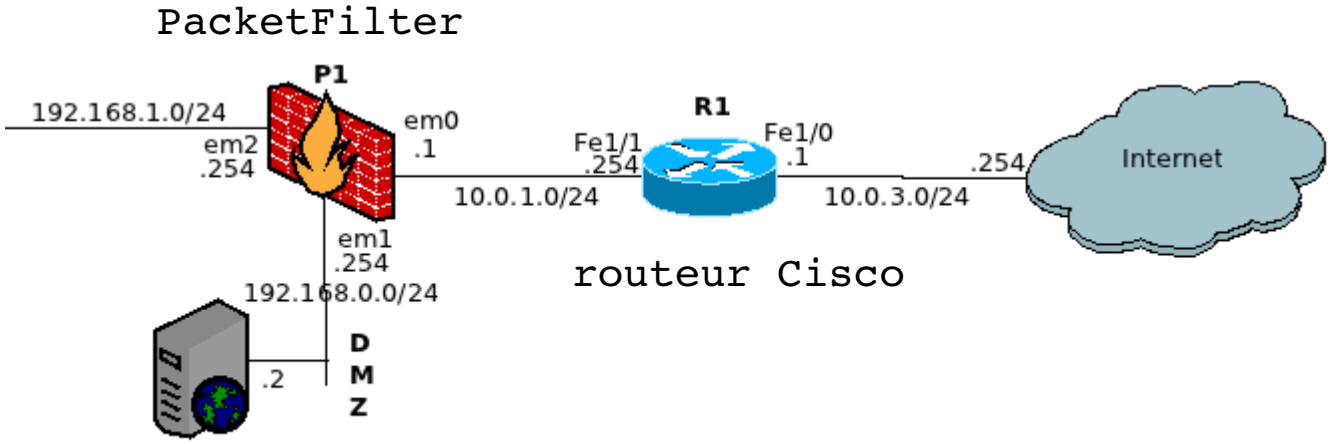
- Solution : émuler le comportement des équipements (filtrage, routage).



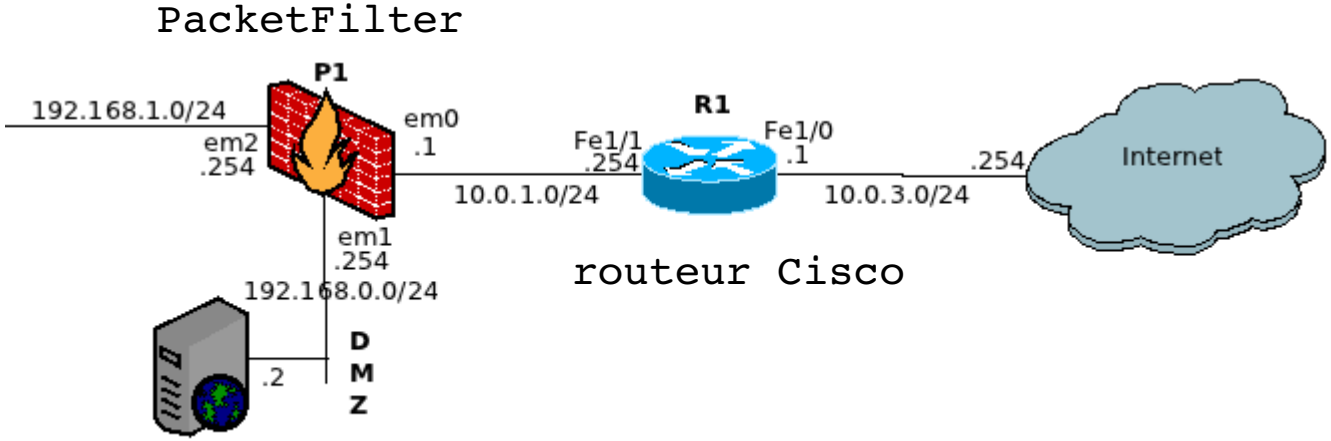
Configuration

- Un fichier de configuration générale ;
- Un fichier de configuration par équipement à émuler ;
- Utilise les fichiers de configurations natifs des équipements (Cisco PIX, routeur Cisco, PacketFilter).

Exemple


PacketFilter
P1
192.168.1.0/24
em2
.254
em0
.1
em1
.254
192.168.0.0/24
.2
D
M
Z
10.0.1.0/24
R1
Fe1/1
.254
Fe1/0
.1
10.0.3.0/24
.254
Internet
routeur Cisco

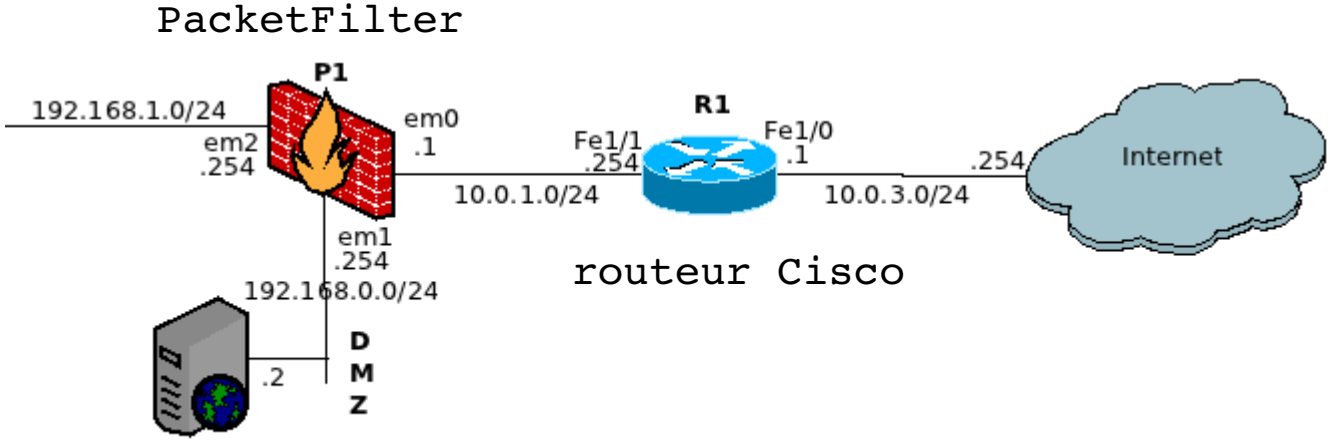
Shell



- Outil texte, interaction via un interpréteur de commande (shell).

```
lsfw> topology
```

Shell



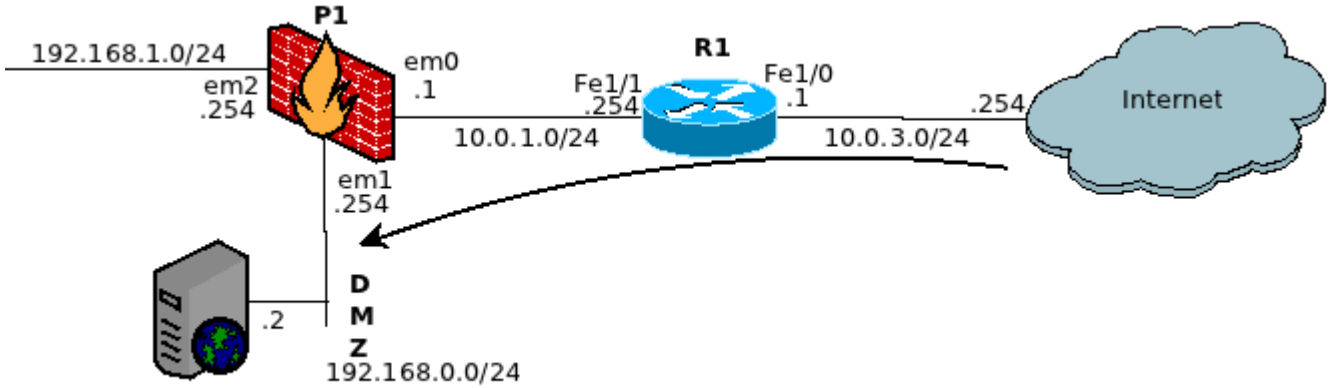
- Outil texte, interaction via un interpréteur de commande (shell).

```
lsfw> topology
10.0.1.0/24 {R1(FastEthernet1/1 - 10.0.1.254), P1(em0 - 10.0.1.1)}
10.0.2.0/24 {R1(FastEthernet1/2 - 10.0.2.254)}
10.0.3.0/24 {R1(FastEthernet1/0 - 10.0.3.1)}
192.168.0.0/24 {P1(em1 - 192.168.0.254)}
192.168.1.0/24 {P1(em2 - 192.168.1.254)}
```

probe 1.2.3.4 192.168.0.0/24

- Sondage source vers destination.

```
lsfw> probe 1.2.3.4 192.168.0.0/24
```



probe 1.2.3.4 192.168.0.0/24

- Chemin suivi **équipement** par **équipement**

```
-------- Routed probes --------
Path:
On: R1 (cisco router #1)
    FastEthernet1/0 (internet)
        interface IP: 10.0.3.1 network: 10.0.3.0/24
    FastEthernet1/1 (INTERNAL)
        interface IP: 10.0.1.254 network: 10.0.1.0/24
        nexthop: 10.0.1.1
On: P1 (firewall #1)
    em0 (em0)
        interface IP: 10.0.1.1 network: 10.0.1.0/24
    em1 (em1)
        interface IP: 192.168.0.254 network: 192.168.0.0/24
        nexthop: 192.168.0.0/24
```

probe 1.2.3.4 192.168.0.0/24

- Et chemin suivi **interface** par **interface**

```
-------- Routed probes --------
Path:
On: R1 (cisco router #1)
    FastEthernet1/0 (internet)
        interface IP: 10.0.3.1 network: 10.0.3.0/24
    FastEthernet1/1 (INTERNAL)
        interface IP: 10.0.1.254 network: 10.0.1.0/24
        nexthop: 10.0.1.1
On: P1 (firewall #1)
    em0 (em0)
        interface IP: 10.0.1.1 network: 10.0.1.0/24
    em1 (em1)
        interface IP: 192.168.0.254 network: 192.168.0.0/24
        nexthop: 192.168.0.0/24
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur R1)

```
R1 (cisco router #1)
Matching ACL on input: FastEthernet1/0 (internet)
  ACCEPT  r1-conf #25: [INTERNET_IN] permit icmp any any
   DENY  r1-conf #26: [INTERNET_IN] deny udp any any range 135 139
   DENY  r1-conf #27: [INTERNET_IN] deny tcp any any range 135 139
...
  ACCEPT  r1-conf #34: [INTERNET_IN] permit ip any any
   DENY  [INTERNET_IN] *** implicit deny ***
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur R1)

```
R1 (cisco router #1)
Matching ACL on input: FastEthernet1/0 (internet)
  ACCEPT  r1-conf #25: [INTERNET_IN] permit icmp any any
   DENY  r1-conf #26: [INTERNET_IN] deny udp any any range 135 139
   DENY  r1-conf #27: [INTERNET_IN] deny tcp any any range 135 139
...
  ACCEPT  r1-conf #34: [INTERNET_IN] permit ip any any
   DENY  [INTERNET_IN] *** implicit deny ***
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur R1)

```
R1 (cisco router #1)
Matching ACL on input: FastEthernet1/0 (internet)
  ACCEPT  r1-conf #25: [INTERNET_IN] permit icmp any any
   DENY  r1-conf #26: [INTERNET_IN] deny udp any any range 135 139
   DENY  r1-conf #27: [INTERNET_IN] deny tcp any any range 135 139
...
  ACCEPT  r1-conf #34: [INTERNET_IN] permit ip any any
   DENY  [INTERNET_IN] *** implicit deny ***
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur R1)

```
R1 (cisco router #1)
Matching ACL on input: FastEthernet1/0 (internet)
  ACCEPT  r1-conf #25: [INTERNET_IN] permit icmp any any
   DENY  r1-conf #26: [INTERNET_IN] deny udp any any range 135 139
   DENY  r1-conf #27: [INTERNET_IN] deny tcp any any range 135 139
...
  ACCEPT  r1-conf #34: [INTERNET_IN] permit ip any any
   DENY  [INTERNET_IN] *** implicit deny ***
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur P1)

```
P1 (firewall #1)
Matching ACL on input: em0 (em0)
   DENY  pf.conf #26: block all
  ACCEPT pf.conf #27: pass quick inet proto icmp from any to
any
  MAY ACCEPT  pf.conf #33: pass proto tcp from any to
192.168.0.2 port { http, https }
```

probe 1.2.3.4 192.168.0.0/24

- Règles de filtrage qui correspondent à la demande (sur P1)

```
P1 (firewall #1)
Matching ACL on input: em0 (em0)
   DENY  pf.conf #26: block all
  ACCEPT pf.conf #27: pass quick inet proto icmp from any to
any
  MAY ACCEPT  pf.conf #33: pass proto tcp from any to
192.168.0.2 port { http, https }
```

La règle correspond partiellement : on teste vers 192.168.0.0/24 ce qui inclus le serveur http, d'où le « peut-être accepté (MAY ACCEPT) »

probe 1.2.3.4 192.168.0.0/24

- Résultat global du filtrage et du routage

```
Global ACL result is: ACCEPT
Global routing result is: ROUTED
```

- ACCEPT !?

Non il n'y a pas d'erreur :

```
P1 (firewall #1)
Matching ACL on input: em0 (em0)
   DENY  pf.conf #26: block all
  ACCEPT pf.conf #27: pass quick inet proto icmp from any to
any
```

probe 1.2.3.4 192.168.0.2 tcp dyn:http

- Si on précise la demande, le résultat est plus précis

```
lsfw> probe 1.2.3.4 192.168.0.2 tcp dyn:http

Résultat sur P1 :

P1 (firewall #1)
Matching ACL on input: em0 (em0)
   DENY   pf.conf #26: block all
  ACCEPT  pf.conf #33: pass proto tcp from any to 192.168.0.2
port { http, https }
```

Pour finir

- Possibilité de dérouler des suites de tests (pour des tests de non régression ou de contrôle de la politique de sécurité) ;

- Quelques autres commandes utiles (topologie, références croisées d'adresses IP) ;

- Application utilisée depuis un an sur deux sites (Université de Rennes 1, IRISA Rennes).

- Prévue dès le début pour être diffusée en libre (double licence esup-Portail/BSD), disponible via https://listes.cru.fr/wiki/jtacl/ ;

- Assez complète : 6 mois de développement, ~40 000 lignes de code (Java).

Questions ?

- Des questions ?

Merci !