Lists, Tables, and Macros

Networks Three

Otago Polytechnic
Dunedin, New Zealand

Firewall rules

Last time we started creating firewall rules. They include

- an action (`block` or `drop`;
- specifications for matching packets, such as
  - source or destination IP address
  - source or destination port
  - transport protocol
  - network interface
- Example: `pass in on em0 to 172.20.44.8 to port 80`

Multiple rules

In many situations you may find yourself writing pretty repetitive rules.

```
pass in on em0 proto tcp to 192.168.5.5 port 22
pass in on em0 proto tcp to 192.168.5.5 port 25
pass in on em0 proto tcp to 192.168.5.5 port 80
pass in on em0 proto tcp to 192.168.5.5 port 443
```

You can condense a rule set like this with a list.

```
pass in on em0 proto tcp to 192.168.5.5 port { 22 25 80 443 }
```

PF just expands your rule to four rules as they are shown above.

List uses

You can use a list anywhere where you want to insert mulitple values in a rule set.

```
block out from { 192.168.6.211 10.115.8.20 } port www

pass in proto { udp tcp } to 172.18.18.6 port 53
```

List expansion

Rules with lists are expanded out to multiple rules. You need to be careful to avoid unintended consequences.

```
pass in on fxp0 from { 10.0.0.0/8, !10.1.2.3 }
```

Expands to

```
pass in on fxp0 from 10.0.0.0/8
pass in on fxp0 from !10.1.2.3
```

Which is almost certainly not what was intended.

Macros

Macros are user-defined variables.

```
allowed_ports = "{ 22 80 443 }"
web_server = "10.10.1.5"
pass in to $web_server port $allowed_ports
```

Macros in lists

You can use macros in lists.

```
malicious_ips = "{ 192.168.15.7 172.21.233.18 }"
marketing = "10.40.1.0/24"
block all from { $malicious_ips $marketing }
```

Tables

- Tables hold groups of IP addresses.
- Tables lookup are faster and use less resources than lists.
- Tables can be modified on the fly.
- Table syntax handles negations nicely (unlike lists).

Declaring tables

```
table <office_net> { 192.168.10.0/24 }
table <allowed_out> const { 10.10.0.0/16 !10.10.5.5 }
table <spammers> persist file "/etc/spammers"
```

Manipulating tables

Tables not declared as `const` can be modified with `pfctl`

```
pfctl -t spammers -T add 203.0.113.0/24

pfctl -t spammers -T show

pfctl -t spammers -T delete 203.0.113.0/24
```