

IPTables — Part 1

Open Source Security Tools for Information Technology Professionals

School of Professional Studies (SPS)

The City University of New York (CUNY)

Aron Trauring
Adjunct Professor
CEO, Zoteca

Class 5 October 24th, 2005

Netfilter

- Linux kernel built-in stateful packet-filtering firewall
- Controlled in user space by `iptables` command

Netfilter Components

- Tables which contain
- Chains which contain
- Rules — criteria to match traffic and apply specific set of actions

Standard Tables

- `filter` — default table for filtering traffic
- `nat` — network address translation rules
- `mangle` — packet alteration functions

Standard Chains (for `filter`)

- First evaluates which chain a packet is destined for
- INPUT — packet coming into host on a network interface
- OUTPUT — packet generated by host going out to network
- FORWARD — entered host but ultimately destined for elsewhere (e.g. on perimeter firewalls)

Rules

- iptables [command] [rule specifications] [extensions]

Commands

- `-A [chain]` — add rule to the end of the chain
- `-I [chain rulenum]` — add rule to position rulenum in chain
- `-R [chain rulenum]` — replaces rule
- `-D [chain [rulenum]]` — replaces rule
- `-L [chain]` — list all rules
- `-F` — flush all rules; good to start
- `-P [chain]` — set default policy for chain

Rule Specifications

- `-i / -o` — ethernet interface
- `-p` — protocol (tcp, udp, icmp, number)
- socket
- state inspection
- policy

Socket

- `-s` — source IP address
- `--sport` — source port
- `-d` — destination IP address
- `--dport` — destination port
- IP address can be any valid address or range including masks
- Port can be number or name (as defined in `/etc/services`)

State Inspection

- `-m state` — enable state module
- `--state [state]`
- NEW — new connection
- ESTABLISHED — existing connection in the process of sending data
- RELATED — connection used to facilitate another connection
- INVALID — connection having problems
- refines rules on connections to further restrict traffic

Policies

- `-j policy`
- ACCEPT — allow
- DROP — discards without notification to sender
- REJECT — discards but sends ICMP notification
- DROP is harsh (remote device needs to timeout) but more secure (less information provided)
- LOG — log traffic

Example Commands

- `iptables -A INPUT -i eth0 -p tcp --dport http -d 192.168 -j ACCEPT`
- `iptables -A OUTPUT -o eth0 -p tcp --dport http -j ACCEPT`
- `iptables -F INPUT`
- `iptables -L`

Example Basic Firewall — Initial Setup

- Flush and set basic policies
- Allow loopback to be freely useable
- Setup logging (log rules must come before others or will never be executed)

```
iptables -F
```

```
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A OUTPUT -o lo -j ACCEPT
```

```
iptables -A INPUT -i eth0 -j LOG --log-prefix "IPT_INPUT"  
"  
--log-level warning  
  
iptables -A OUTPUT -o eth0 -j LOG --log-prefix "IPT_OUTPUT"  
"  
--log-level warning
```

Example Basic Firewall — HTTP Server

- allow all incoming HTTP
- restrict outgoing to established connections

```
iptables -A INPUT -i eth0 -p tcp --dport http -d 192.168  
-m state  
--state NEW,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p tcp --sport http -d  
192.168.0.1 -m state  
--state ESTABLISHED -j ACCEPT
```

Example Basic Firewall — DNS Queries

- allow establishing new connections for output only
- restrict to specific DNS servers using `-s/-d` flags

```
iptables -A INPUT -i eth0 -p udp --sport domain -s  
192.168.0/24 -m state  
--state ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p udp --dport http -d  
192.168.0/24 -m state  
--state NEW,ESTABLISHED -j ACCEPT
```

Example Basic Firewall — Remote SSH administration

- allow establishing new connections for output only
- restrict to specific DNS servers using `-s/-d` flags

```
iptables -A INPUT -i eth0 -p tcp -d 192.168.2.11 --sport  
ssh
```

```
-m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -o eth0 -p tcp -s 192.168.2.11  
--dport ssh
```

```
-m state --state NEW,ESTABLISHED -j ACCEPT
```

ICMP (Internet Control Message Protocol) Traffic

- Various message types
- `ping` — `echo request (8)` followed by an `echo reply (0)`
- `traceroute` — `destination unreachable (3)` — host down or declining ICMP
- `traceroute` — `time exceeded (11)` — used for mapping

ICMP Attacks

- Flood attacks — storm of pings overwhelm system resulting in Denial of Service (DoS)
- Smurf attacks — forged ICMP packets sent to network broadcast addresses; results in DoS of forged recipient

- Ping of Death — ICMP packet larger than maximum IP packet size causes system to crash
- Nuke attack — ICMP packet contains information system can't handle causing system to crash

ICMP Rules

- Allow outbound echo request and inbound echo reply for ping from host.
- Allow destination unreachable and time exceeded inbound for traceroute.

ICMP Rules — Set Up

- Create new chains and redirect traffic to those chains

```
iptables -N ICMP_IN
```

```
iptables -N ICMP_OUT
```

```
iptables -A INPUT -p icmp -j ICMP_IN
```

```
iptables -A OUTPUT -p icmp -j ICMP_OUT
```

ICMP Rules — Inbound

- Drop inbound ping request
- Allow inbound echo reply, destination unreachable and time exceeded

```
iptables -A ICMP_IN -p icmp --icmp-type echo-request  
-j DROP
```

```
iptables -A ICMP_IN -i eth0 -p icmp --icmp-type 0  
-m state  
--state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A ICMP_IN -i eth0 -p icmp --icmp-type 3  
-m state  
--state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A ICMP_IN -i eth0 -p icmp --icmp-type 11  
-m state  
--state ESTABLISHED,RELATED -j ACCEPT
```

ICMP Rules — Inbound

- Create a new log chain
- Redirect rest of incoming icmp traffic to log chain
- Drop and log this traffic

```
iptables -N LOG_DROP
```

```
iptables -A ICMP_IN -i eth0 -p icmp -j LOG_DROP
```

```
iptables -A LOG_DROP -i eth0 -p icmp -j LOG --log-prefix  
"IPT_ICMP_IN "
```

```
iptables -A LOG_DROP -i eth0 -p icmp -j DROP
```

ICMP Rules — Outbound

- Allow outbound `echo` request
- Log and drop all other ICMP outbound

```
iptables -A ICMP_OUT -o eth0 -p icmp --icmp-type 8  
-m state  
--state NEW -j ACCEPT
```

```
iptables -A ICMP_OUT -o eth0 -p icmp -j LOG_DROP
```

```
iptables -A LOG_DROP -o eth0 -p icmp -j LOG --log-prefix  
"IPT_ICMP_OUT "
```

```
iptables -A LOG_DROP -o eth0 -p icmp -j DROP
```