ebtables Command Overview

ebtables is a user space tool. It is used to setup and maintain the tables of Ethernet frame rules in the MB/1520-100 Linux kernel. Together with ***brctl*** (Ethernet Bridging Tool) it allows you to build a OSI layer 2 (Data Link Layer) bridge firewall within seconds.

The user interface to ***ebtables*** is command line-based. When connected via a RS232 serial line or a Telnet session to the MB/1520-100 you can interactively enter commands and see the results. The general format of an ***ebtables*** statement is:

```
ebtables command rule-specifications extensions
```

where **`command`**, **`rule-specifications`** and **`extensions`** are one or more of the valid options. Table 1 lists the ***ebtables*** commands. Table 2 contains the ***ebtables*** rule specifications.

Command	**Function**
-A, --append	Append a rule to the end of the selected chain.
--atomic-commit	Replace the kernel table data with the data contained in the specified file.
--atomic-file	Let the command operate on the specified file.
--atomic-file -Z	Zero counters within a file.
--atomic-init	Copy the kernel's initial data of the table to the specified file.
--atomic-save	Copy the kernel's current data of the table to the specified file.
-D, --delete	Delete the specified rule from the selected chain.
-E, --rename-chain	Rename the specified chain to a new name.
-F, --flush	Flush the selected chain.
-h, --help	Give a brief description of the command syntax.
-I, --insert	Insert the specified rule into the selected chain at the specified rule number.
--init-table	Replace the current table data by the initial table data.
-j, --jump	The target of the rule.
-L, --list	List all rules in the selected chain.
-M, --modprobe	Load missing kernel modules.
-N, --new-chain	Create a new user-defined chain with the given name.
-P, --policy	Set the policy for the chain to the given target.
-V, --version	Show the version of the ebtables user space program.
-X, --delete-chain	Delete the specified user-defined chain.
-Z, --zero	Set the counters of the selected chain to zero.

Table 1: The ***ebtables*** commands

The following command line arguments make up a rule specification (as used in the add and delete commands). A "!" option before the specification inverts the test for that specification.

Parameter	**Function**
-p, --protocol [!] *protocol*	The protocol that was responsible for creating the frame.
-i, --in-interface [!] *name*	The interface via which a frame is received.
--logical-in [!] *name*	The (logical) bridge interface via which a frame is received.
-o, --out-interface [!] *name*	The interface via which a frame is going to be sent.
--logical-out [!] *name*	The (logical) bridge interface via which a frame is going to be sent.
-s, --source [!] *address*[/*mask*]	The source mac address.
-d, --destination [!] *address*[/*mask*]	The destination mac address.

Table 2: The ***ebtables*** rule specifications

The following sample code shows a ***ebtables***-based basic filter configuration which will only let frames made by the protocols IP version 4 and ARP through. Also, the network has some old machines that use the protocol field of the Ethernet frame as a length field (they use the Ethernet 802.2 or 802.3 protocol). There was no reason not to let those machines through, more precisely: there was a reason to let them through. So, those frames, with protocol LENGTH denoting that it's really a length field, are accepted. Of course one could filter on the MAC addresses of those old machines so no other machine can use the old Ethernet 802.2 or 802.3 protocol. All other frames get logged and dropped. This logging consists of the protocol number, the MAC addresses, the IP/ARP info (if it's an IP/ARP packet of course) and the in and out interfaces.

```
ebtables -P FORWARD DROP
ebtables -A FORWARD -p IPv4 -j ACCEPT
ebtables -A FORWARD -p ARP -j ACCEPT
ebtables -A FORWARD -p LENGTH -j ACCEPT
ebtables -A FORWARD  --log-level info --log-ip --log-prefix EBFW
ebtables -P INPUT DROP
ebtables -A INPUT -p IPv4 -j ACCEPT
ebtables -A INPUT -p ARP -j ACCEPT
ebtables -A INPUT -p LENGTH -j ACCEPT
ebtables -A INPUT --log-level info --log-ip --log-prefix EBFW
ebtables -P OUTPUT DROP
ebtables -A OUTPUT -p IPv4 -j ACCEPT
ebtables -A OUTPUT -p ARP -j ACCEPT
ebtables -A OUTPUT -p LENGTH -j ACCEPT
ebtables -A OUTPUT --log-level info --log-ip --log-arp --log-prefix EBFW -j DROP
```

Listing 1: ***ebtables*** sample

ebtables comes with an internal packet filter for different protocol frame formats. This filter is the default table for rule specifications (i.e. with the append command -A).

The ***ebtables*** filter contains three built-in chains: **INPUT** (for frames destined for the bridge itself), **OUTPUT** (for locally-generated frames) and **FORWARD** (for frames being bridged). Figure 1 shows the architecture. Each append command specifies one chain.

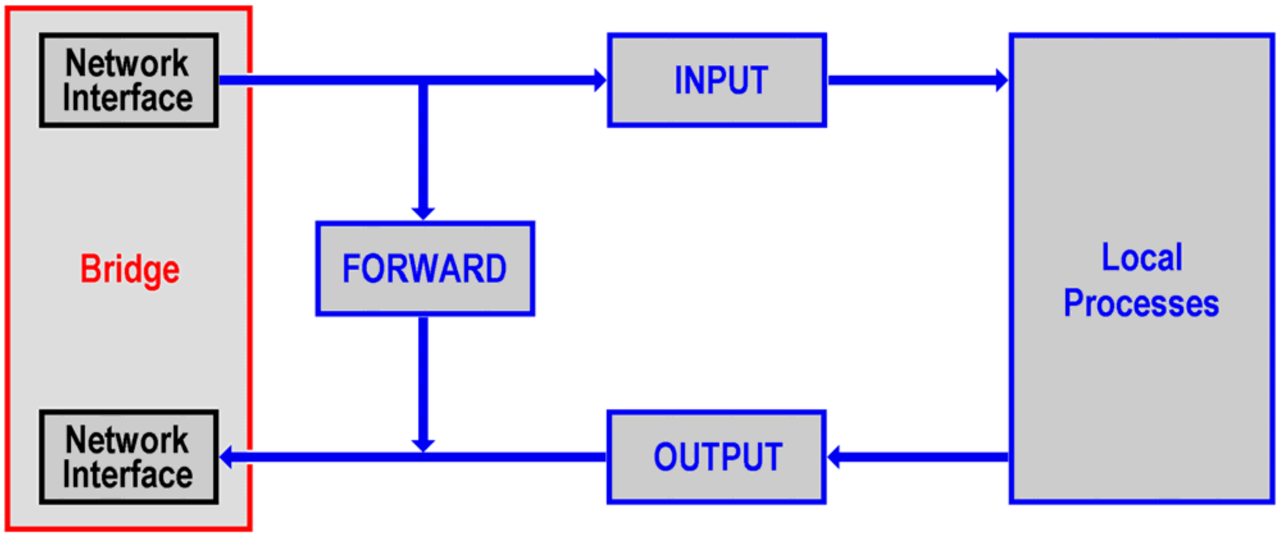

Figure 1: ***ebtables*** function diagram