NAME

mac_biba - Biba data integrity policy

SYNOPSIS

To compile Biba into your kernel, place the following lines in your kernel configuration file:

```
options MAC
options MAC_BIBA
```

Alternately, to load the Biba module at boot time, place the following line in your kernel configuration file:

```
options MAC
```

and in loader.conf(5):

```
mac_biba_load="YES"
```

DESCRIPTION

The **mac_biba** policy module implements the Biba integrity model, which protects the integrity of system objects and subjects by means of a strict information flow policy. In Biba, all system subjects and objects are assigned integrity labels, made up of hierarchal grades, and non-hierarchal components. Together, these label elements permit all labels to be placed in a partial order, with information flow protections based on a dominance operator describing the order. The hierarchal grade field is expressed as a value between 0 and 65535, with higher values reflecting higher integrity. The non-hierarchal compartment field is expressed as a set of up to 256 components, numbered from 0 to 255. A complete label consists of both hierarchal and non-hierarchal elements.

Three special label values exist:

Label	**Comparison**
biba/low	lower than all other labels
biba/equal	equal to all other labels
biba/high	higher than all other labels

The "biba/high" label is assigned to system objects which affect the integrity of the system as a whole. The "biba/equal" label may be used to indicate that a particular subject or object is exempt from the Biba protections. These special label values are not specified as containing any compartments, although in a label comparison, "biba/high" appears to contain all compartments, "biba/equal" the same compartments as the other label to which it is being compared, and "biba/low" none.

In general, Biba access control takes the following model:

- A subject at the same integrity level as an object may both read from and write to the object as though Biba protections were not in place.
- A subject at a higher integrity level than an object may write to the object, but not read the object.
- A subject at a lower integrity level than an object may read the object, but not write to the object.
- If the subject and object labels may not be compared in the partial order, all access is restricted.

These rules prevent subjects of lower integrity from influencing the behavior of higher integrity subjects by preventing the flow of information, and hence control, from allowing low integrity subjects to modify either a high integrity object or high integrity subjects acting on those objects. Biba integrity policies may be appropriate in a number of environments, both from the perspective of preventing corruption of the operating system, and corruption of user data if marked as higher integrity than the attacker. In traditional trusted operating systems, the Biba integrity model is used to protect the Trusted Code Base (TCB).

The Biba integrity model is similar to mac_lomac(4), with the exception that LOMAC permits access by a higher integrity subject to a lower integrity object, but downgrades the integrity level of the subject to prevent integrity rules from being violated. Biba is a fixed label policy in that all subject and object label changes are explicit, whereas LOMAC is a floating label policy.

The Biba integrity model is also similar to mac_mls(4), with the exception that the dominance operator and access rules are reversed, preventing the downward flow of information rather than the upward flow of information. Multi-Level Security (MLS) protects the confidentiality, rather than the integrity, of subjects and objects.

Label Format

Almost all system objects are tagged with an effective, active label element, reflecting the integrity of the object, or integrity of the data contained in the object. In general, objects labels are represented in the following form:

biba/*grade*:*compartments*

For example:

biba/10:2+3+6
biba/low

Subject labels consist of three label elements: an effective (active) label, as well as a range of available labels. This range is represented using two ordered Biba label elements, and when set on a process, permits the process to change its active label to any label of greater or equal integrity to the low end of the range, and lesser or equal integrity to the high end of the range. In general, subject labels are represented in the following form:

biba/*effectivegrade*:*effectivecompartments*(*lograde*:*locompartments*-*higrade*:*hicompartments*)

For example:

biba/10:2+3+6(5:2+3-20:2+3+4+5+6)
biba/high(low-high)

Valid ranged labels must meet the following requirement regarding their elements:

rangehigh >= *effective* >= *rangelow*

One class of objects with ranges currently exists, the network interface. In the case of the network interface, the effective label element references the default label for packets received over the interface, and the range represents the range of acceptable labels of packets to be transmitted over the interface.

Runtime Configuration

The following sysctl(8) MIBs are available for fine-tuning the enforcement of this MAC policy.

security.mac.biba.enabled Enables enforcement of the Biba integrity policy. (Default: 1).

security.mac.biba.ptys_equal Label pty(4)s as "biba/equal" upon creation. (Default: 0).

security.mac.biba.revocation_enabled
Revoke access to objects if the label is changed to dominate the subject. (Default: 0).

SEE ALSO

mac(4), mac_bsdextended(4), mac_ifoff(4), mac_lomac(4), mac_mls(4), mac_none(4), mac_partition(4), mac_portacl(4), mac_seeotheruids(4), mac_test(4), maclabel(7), mac(9)

HISTORY

The **mac_biba** policy module first appeared in FreeBSD 5.0 and was developed by the TrustedBSD Project.

AUTHORS

This software was contributed to the FreeBSD Project by Network Associates Labs, the Security Research Division of Network Associates Inc. under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"), as part of the DARPA CHATS research program.