NAME

mac_lomac - Low-watermark Mandatory Access Control data integrity policy

SYNOPSIS

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

```
options MAC
options MAC_LOMAC
```

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

```
options MAC
```

and in loader.conf(5):

```
mac_lomac_load="YES"
```

DESCRIPTION

The **mac_lomac** policy module implements the LOMAC integrity model, which protects the integrity of system objects and subjects by means of an information flow policy coupled with the subject demotion via floating labels. In LOMAC, all system subjects and objects are assigned integrity labels, made up of one or more hierarchical grades, depending on their types. Together, these label elements permit all labels to be placed in a partial order, with information flow protections and demotion decisions based on a dominance operator describing the order. The hierarchal grade field or fields are expressed as a value between 0 and 65535, with higher values reflecting higher integrity.

Three special label component values exist:

Label	Comparison
low	dominated by all other labels
equal	equal to all other labels
high	dominates all other labels

The "high" label is assigned to system objects which affect the integrity of the system as a whole. The "equal" label may be used to indicate that a particular subject or object is exempt from the LOMAC protections. For example, a label of "lomac/equal(equal-equal)" might be used on a subject which is to be used to administratively relabel anything on the system.

Almost all system objects are tagged with a single, active label element, reflecting the integrity of the

object, or integrity of the data contained in the object. File system objects may contain an additional auxiliary label which determines the inherited integrity level for new files created in a directory or the alternate label assumed by the subject upon execution of an executable. In general, objects labels are represented in the following form:

lomac/*grade*[*auxgrade*]

For example:

lomac/10[2]
lomac/low

Subject labels consist of three label elements: a single (active) label, as well as a range of available labels. This range is represented using two ordered LOMAC 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:

lomac/*singlegrade*(*lograde*-*higrade*)

Modification of objects is restricted to access via the following comparison:

subject::*higrade* >= *target-object*::*grade*

Modification of subjects is the same, as the target subject's single grade is the only element taken into comparison.

Demotion of a subject occurs when the following comparison is true:

subject::*singlegrade* > *object*::*grade*

When demotion occurs, the subject's *singlegrade* and *higrade* are reduced to the object's grade, as well as the *lograde* if necessary. When the demotion occurs, in addition to the permission of the subject being reduced, shared mmap(2) objects which it has opened in its memory space may be revoked according to the following sysctl(3) variables:

- *security.mac.lomac.revocation_enabled*
- *security.mac.enforce_vm*
- *security.mac.mmap_revocation*
- *security.mac.mmap_revocation_via_cow*

Upon execution of a file, if the executable has an auxiliary label, and that label is within the current range of *lograde-higrade*, it will be assumed by the subject immediately. After this, demotion is performed just as with any other read operation, with the executable as the target. Through the use of auxiliary labels, programs may be initially executed at a lower effective integrity level, while retaining the ability to raise it again.

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. LOMAC 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.

The LOMAC security model is quite similar to that of mac_biba(4) and mac_mls(4) in various ways. More background information on this can be found in their respective man pages.

SEE ALSO

mmap(2), sysctl(3), mac(4), mac_biba(4), mac_bsdextended(4), mac_ddb(4), mac_ifoff(4), mac_mls(4), mac_none(4), mac_partition(4), mac_portacl(4), mac_seeotheruids(4), mac_test(4), mac(9)

HISTORY

The **mac_lomac** 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.