NetLabel Linux Security Module Interface

Paul Moore, paul.moore@hp.com

May 17, 2006

* Overview

NetLabel is a mechanism which can set and retrieve security attributes from network packets. It is intended to be used by LSM developers who want to make use of a common code base for several different packet labeling protocols. The NetLabel security module API is defined in 'include/net/netlabel.h' but a brief overview is given below.

* NetLabel Security Attributes

Since NetLabel supports multiple different packet labeling protocols and LSMs it uses the concept of security attributes to refer to the packet's security labels. The NetLabel security attributes are defined by the 'netlbl_lsm_secattr' structure in the NetLabel header file. Internally the NetLabel subsystem converts the security attributes to and from the correct low-level packet label depending on the NetLabel build time and run time configuration. It is up to the LSM developer to translate the NetLabel security attributes into whatever security identifiers are in use for their particular LSM.

* NetLabel LSM Protocol Operations

These are the functions which allow the LSM developer to manipulate the labels on outgoing packets as well as read the labels on incoming packets. Functions exist to operate both on sockets as well as the sk_buffs directly. These high level functions are translated into low level protocol operations based on how the administrator has configured the NetLabel subsystem.

* NetLabel Label Mapping Cache Operations

Depending on the exact configuration, translation between the network packet label and the internal LSM security identifier can be time consuming. The NetLabel label mapping cache is a caching mechanism which can be used to sidestep much of this overhead once a mapping has been established. Once the LSM has received a packet, used NetLabel to decode its security attributes, and translated the security attributes into a LSM internal identifier the LSM can use the NetLabel caching functions to associate the LSM internal identifier with the network packet's label. This means that in the future when a incoming packet matches a cached value not only are the internal NetLabel translation mechanisms bypassed but the LSM translation mechanisms are bypassed as well which should result in a significant reduction in overhead.