GLOSSARY

This glossary defines terms peculiar to IX. The glossary for the Unix Research System, 10th Edition, which is incorporated by reference, defines certain terms used here: *argument, executable file, file, groupid, inode, kernel, permission, process, stream, superuser, system call, terminal, u-area, umask, userid, utility.*

accept pex indicator a control, set with *privilege* [1], on a stream to permit or deny *pexing* according as the stream is or is not *trusted* [3].

assured path a channel comprising *trusted* streams and processes that is understood to pass information faithfully without tampering or eavesdropping.

audit to record security-related events, such as file accesses, process creation, and exercise of *privilege* [1].

audit mask a bit vector associate with each process to specify the intensity of *auditing.*

bottom see *lattice label.*

capability 1. actual right of a process to exercise a *privilege* [2]; cf. *license.* Process capabilities, which can be relinquished at any time, are determined at *exec*(2), either by intersecting its licenses and the *capabilities* [2] of the file it is executing or by *self-licensing.* 2. potential right of an executable file to exercise privilege.

ceiling a *label* [1], which must dominate the label of any file involved in a system call. Every process and every file system has a ceiling.

constant see *fixity.*

covert channel an information path between untrusted processes that does not obey the *mandatory security policy.* Always of low bandwidth, covert channels usually involve inferences from error returns rather than *data flows.*

data flow explicit transfer of bits from place to place by system calls. Pertinent places are processes, files, directories, inodes, seek pointers, and u-area data, such as process *ceiling,* exit status, umask, userid, and groupid; cf. *covert channel.*

domination a relationship among *labels* [1]. A *lattice label* is said to **dominate** another if and only if the former has one bits in all positions that the latter does. A label with label flag value *yes* dominates and is dominated by any label. A label with *label flag* value *no* does not dominate and is not dominated by **no** or by any lattice label.

downgrade to change, by use of *privilege,* the lattice label of a file to a lattice label that does not *dominate* the previous value.

drop 1. to change the value of a process *label* so that the new value does not *dominate* the old value. A process label can drop only at *exec*(2) with no arguments. 2. to decrease the *ceiling* of a process, as by *drop*(1).

extern a *privilege* [2] that allows the *label* [1] of an open *external medium* to be set away from its quiescent value of **no**.

external medium a file, such as a terminal or magnetic tape, that communicates with the outside world. Because the *mandatory security policy* cannnot automatically be assured on external media, *privilege* [2] is required to initiate input/output thereon.

fixity the degree to which a *label* [1] on a file or process may be changed. The values of fixity are: **loose,** freely changeable to a dominating value; **frozen,** changeable only explicitly by the owner; **rigid,** changeable only with privilege; and **constant,** not changeable.

floor a conventional *lattice label* [1] assigned to a user's shell process at login. The floor is the label of the file `/etc/floor`.

frozen see *fixity.*

label 1. a designation of the *mandatory security* status of a file or process. 2. the representation of a label [1], comprising: *label flag, fixity, lattice label, capabilities* [2], and *licenses* [2].

label flag part of a *label* [2] that tells whether the label's value is a *lattice label,* or one of two special values, *yes* for generally readable and writable data, such as `/dev/null`, or *no* for generally unreadable and unwritable data, such as *external media.*

lattice label a designation of security level, the lattice label comprises 480 bits. Data flow is permitted only if the lattice label of the destination *dominates* the lattice label of the source. Lattice labels of all zeros and all ones are called **bottom** and **top** respectively.

license 1. potential right of a process to exercise a *privilege* [2]. A license can be relinquished at any time and is inherited across *exec*(2). 2. an indicator of *self-licensing* of a file.

log a *privilege* [2] that allows querying and changing the intensity of *auditing.*

log file a special file for *audit* information. A log file can be written regardless of labels and can be read by no process. Audit files are associated with ordinary files by *setlog*(2).

loose see *fixity.*

mandatory security policy rules to govern *data flow* regardless of 'discretionary' user decisions about file permissions. Except on certain actions of *trusted* processes, a security *label* of the destination of any data flow must *dominate* the label of the source. Labels are calculated at every system call and are adjusted as necessary to preserve dominance. cf. *covert channel* and *TCB.*

no a non-*lattice label* that neither dominates nor is dominated by any *label* [1] other than *yes* . Because a file labeled *no* cannot be read or written by any un*trusted* [2] process, it is safe to set a file label to **no**; cf. *extern.*

nochk a *privilege* [2] that allows a process to access a file regardless of *domination.*

pex to assert process-exclusive access to a file. A pipe pexed at one end can be used only if it is also pexed at the other; see *pex*(4).

poison class a file attribute, visible and settable only with *privilege* [1], that forces auditing to at least a specified *poison mask* level when a process mentions the file.

poison mask one of several auxiliary bit vectors that can augemnt the *audit mask.*

privilege 1. mechanism of *capabilities* and *licenses* for controlling deviation from the basic *mandatory security policy* and for administering privilege. 2. one of six distinct classes of privilege: *extern, log, nochk, setlic, setpriv,* and *uarea;* cf. *trusted.*

privilege server the utility *priv*(1), which, following rules in the file *privs*(5), grants *licenses* [1] needed to exercise *privilege.*

rigid see *fixity.*

self-license possession by a file of a *capability* [2] and a corresponding *license* [2]. Self-licensing gives the corresponding *capability* [1] to a process at *exec*(2).

session an interval of running with special rights, usually evidenced by a distinct terminal *label* [1], *ceiling,* or *stream identifier;* see *session*(1).

setlic a *privilege* [2] that allows the *licenses* [1] or *ceiling* of a process to be set arbitrarily.

setpriv a *privilege* [2] that allows changing the *capabilities* [2] and *licenses* [2] of files.

stream identifier a string that is by exercise of *privilege* [1] attached to a stream to describe properties of the stream and its destination; see **FIOGSRC** and **FIOSSRC** in *stream*(4).

TCB, trusted computing base the kernel, *trusted* [1] utilities, critical data for these utilities, and utilities that may be used to process files in the TCB. Faithfulness to the *mandatory security policy* depends on the correctness of the TCB.

top see *lattice label.*

trusted 1. having some *capability* or *license;* said of a file, especially an executable file. The only way a trusted file can be modified is to change its privileges with capability *setpriv.* 2. having some capability; said of a process. Superuser processes are not necessarily trusted. 3. understood to be immune to tampering or eavesdropping, said of a stream associated with an *external medium;* cf. *assured path.*

trusted computing base Same as *TCB.*

uarea a *privilege* [2] that allows changing userid, groupid, and logname in the per-process u-area. The privilege is required lest these items, being both readable and writable by untrusted processes, provide a means to violate the *mandatory security policy.* The permission mask (umask), and the process *ceiling* are protected by other means; see *exec*(2) and *setplab*(2).

yes a non-*lattice label* that dominates and is dominated by any *label* [1]. A file labeled **yes** can be read or written by any un*trusted* [2] process.