Massachusetts Institute of Technology
6.915: Network and Computer Security
Professor Ronald L. Rivest

Handout 24
October 31, 1995

Problem Set 5 Solutions

Problem 5-1. Auditing

There are at least 3 single points of failures:

1. the `open` routine itself. If somebody could remove the audit calls then the audit scheme would be useless.
2. the auditing routine of course
3. the audit log, since data stored in it could be modified.

Also look at David Yang's solution (attached) for other possible points of failure.

Problem 5-2. User Profiles

Here is a list of the most common answers among the students:

Very important: super user activity, what files accessed (accesses to /etc/passwd for example), attempts to use/modify expired tickets, kerberos ticket file accesses, access on use, editor used, shell used (shell commands used), remote sites accessed, origin of remote accesses, failed login attempts, file execution denied.

Somewhat important: cpu usage, Which applications used (editor, web, zephyr tools), permissions changed, user ids accessed, frequency of logins, number of files deleted and/or created, idleness.

Not very important: which lockers attached, time of logins, length of session, temporary file usage, system errors, printer usage, disk usage.

Problem 5-3. Distributed Intrusion Detection

Sompe people found the question a little ambiguous, in the case when the log is stored elsewhere: did we mean a single dedicated host for remote log storage or a "distributed" mechanism, in which say computer A's logs are stored in computer B's, and B's at D's, etc.

We meant the second objective since the first provide for a single point of failure for the whole distributed system. However there are advantages to use the other alternative as well (it may be easier to secure only one repository of logs.)

Also people were confused by what is meant by offline transmission. Some people thought it involved manual interaction (and coulnd't be automated), and hence an adversary coulnd't modify logs if they were transferred manually. We did not mean that. By offline we basically meant at some convenient time (e.g. at night time when load is low) but still over the network.

The main answers were: **Storing Logs on the same host:**
Pros: simple, easy to analyze, faster detection, load of network unaffected.
Cons: intruder can change log,

Storing Logs on a different host:
Pro: Intruder would have to compromise another host if he wanted to clean his trail. If the remote host is a single one it may be easier to secure one main host, and have just secure transfer from all other networked hosts to this one.
Con: increases network load, need a means for transferring (a protocol that preserves integrity).

Online Transmission:
Pro: earlier detection
Con: network load

Offline Transmission: *Pro:* lower network load
Con: if period is too long, intruder can come in and modify log.

Problem 5-4. Access Control

If capabilities are held by the OS for all users then one way of solving this problem would just be A informing the OS to transfer her capability to B.

Suppose now that users hold their capabilities. The capability should be unforgeable, let us assume that such property is achieved by use of a MAC. A capability is going to be a pair (ticket,MAC) where the ticket specify what and whom the capability is for and the MAC guarantees authenticity,

Let k_U be a shared secret key between user U and the OS. The OS will issue capabilities to perform operations. For example if the OS wants to allow user U to perform operation O the issued capability will look like

$$C_{perf}(U,0) = (T_{perf}(U,O), MAC_{k_U}(T_{perf}(U,O)))$$

Users will issue capabilities used to transfer OS capabilities. If user U wants to transfer a ticket to user U' she will issue him an accompanying ticket

$$C_{trans}(U,U',O) = (T_{trans}(U,U',T_{perf}U,O), MAC_{k_U}(T_{trans}))$$

which basically specifies that U' can use ticket T_{perf} even if it was given to U. So the transferred capability will look like

$$(C_{perf}(U,O), C_{trans}(U,U',O))$$

The OS will accept two kind of capabilitites. If user U wants to perform operation O she has to present to the OS either $C_{perf}(U,O)$ or $(C_{perf}(U'',O), C_{trans}(U'',U,O))$ for some U''.