LTS/OnGuard

Frequently Asked Questions

If your question has not been answered here, please contact us at assist@kuritsolutions.com

General Questions

Q: Does LTS/OnGuard modify VOS?

A: No. LTS/OnGuard uses existing VOS facilities to implement its features.

Q: I have certain users who must have unimpeded privileged access to VOS. Is this still possible if LTS/OnGuard is installed on the system?

A: Yes. LTS/OnGuard doesn't change the user environment unless you choose to do so, which you can do on a user-by-user basis. LTS/OnGuard uses library paths to make its utilities and substitute commands available to users. Privileged users don't need to use the LTS/OnGuard command proxy facilities.

Q: Does LTS/OnGuard require a systems administrator to run it?

A: No. LTS/OnGuard is designed to be run by a security officer or administrator, who may not be familiar with the VOS systems environment. The audit reports do not assume that the user understands the jargon of VOS. It does, however, require a systems administrator to install it.

Q: How does LTS/OnGuard work?

A: LTS/OnGuard has a set of front-end macros which become part of the users' library paths. These front-ends fully mimic the look and feel of the standard VOS commands. When a user desires to use a command, s/he (invisibly) uses the LTS/OnGuard equivalent, which uses our proxy servers with full control and auditability.

Q: What restricts a user from bypassing LTS/OnGuard?

A: LTS/OnGuard is used by non-privileged users who need to execute a privileged command under access and audit constraints. By removing the privileged attribute for the user's registration profile, that user is forced to use LTS/OnGuard if s/he wants to run the command.

Q: Can I control and audit other commands or jobs by using LTS/OnGuard?

A: Yes. By removing access for a particular command or job via the VOS access list facility, you can force the user to use LTS/OnGuard in order to execute the command under control and audit constraints.

Q: Can I control logins and logouts by using LTS/OnGuard?

A: Yes. LTS/OnGuard has very complete user management features, complementing those already in VOS. For example, while VOS has the logout_admin facility, LTS/OnGuard allows you to set up rules for logging out inactive users based on terminal, group, and/or person names.

Q: Does LTS/OnGuard provide real-time alerts?

A: No. LTS/OnGuard does not provide real-time alerts. It provides a wealth of audit reports which may be run at any time, however. If you need real-time alerts, please refer to the VOS command notify_security_violation.

Support Questions

Q: I am seeing LTS processes die or return an error 2452: [e$too_many_user_procs] Maximum processes for this user exceeded. Why am I getting this, and how can it be avoided?

A: Your personal registration on the system has a limit on the number of processes that you can start. With everything you have on the system, including the LTS processes, you are trying to exceed this limit. We recommend that you remove or significantly bump up the limit that is in your registration. Note, too, that users of LTS/OnGuard may start additional processes by using the "–start_session" switch of the execute_privileged_command and execute_as_request commands.

Q: I put a command in the priv cmd table (like display_disk_label.pm), but when I try to use it with epc (epc display_disk_label) I get an "invalid command" message.

A: The command name you pass to the priv cmd environment with epc is checked for an exact match with the table. In this case you should use (epc display_disk_label.pm). By the way, it is good practice to use the correct suffix for external commands in the table to avoid a trojan horse of a command macro to be executed as a privilege user.

Q: How do I determine who is using a terminal?

A: The LTS command audit_login_activity reports on user logins by user, terminal, process, or time.

Q: Certain commands using the privileged command server can take a very long time. This locks out other users from using the privileged commmand server. How can this be avoided?
A: We recommend that the heavy user of the privileged command server do two things:

a) Add the "-start_session" switch to execute_privileged_command so that s/he gets their own copy of the command handler (now the default); and
b) Run the requests in background mode or under batch.

Q: We have been adding commands to the as_rqst_commands.tin table, and we have problems adding the set_word command. When we execute the command line:
!execute_as_request 'set_word sf1 2'
... the following message appears:
LTS/execute_as_request: Invalid command. set_word
... and in the LTS_as_rqst_log.aa-mm-dd :
COMMAND: set_word sf1 2
REASON: set_word not allowed for as_rqst_server.

Why doesn't this work?

A: The as_request_server also checks for certain commands or command types, and always restricts them. Any command with "set_" fails, for example. We feel that these commands open too big a door for mischief, and that they should always be disallowed. Your customer will have to do a privileged session, or, preferably, use the command set_external_variable.

The strings below, if found, are not allowed in a command.

'..' 'iop_monitor' 'ioa_monitor' 'pcp' 'graph_module_usage' 'clone_command' 'interval' 'set_' 'use_'

These commands may be enabled on a site-by-site basis by using set_site_preferences.

'use_block', 'use_iop', 'use_partition', 'set_board_thresholds', 'set_comm_thresholds', 'set_dq_defaults', 'set_lap_trace', 'set_net_timeout' 'set_net_trace',