Platform LSF Security

Platform LSF
Version 8.0
January 2011

Platform Computing

Contents

Platform LSF security considerations

While the default LSF configuration is adequate for most clusters, you should consider the following issues if you want to increase the security of your LSF cluster.

Communications between daemons and commands

Communications between LSF daemons and between LSF commands and daemons are not encrypted. If your LSF clusters are running in an untrusted or unsecured environment, these communications may be susceptible to interception and spoofing attacks. You can enable strict checking of communications to deal with spoofing attacks.

Transmission of Platform LSF commands for remote execution

By default, the following LSF commands make use of remote shell (`rsh`) and remote login (`rlogin`):

- `badmin hstartup`
- `bpeek`
- `lsadmin limstartup`
- `lsadmin resstartup`
- `lsfrestart`
- `lsfshutdown`
- `lsfstartup`
- `lslogin`
- `lsrcp`

`rsh` and `rlogin` may not be suitable for transmission over an insecure network because it is not encrypted. You can configure these LSF commands to use secure shell (`ssh`), which provides encryption when transmitting commands for remote execution.

Access to jobs belonging to other users

All LSF jobs are run under the user ID of the user who submitted the job (unless you are using account mapping). LSF enforces restrictions on job access based on the user ID of the user who is running a command and the user ID associated with the submitted job.

All LSF users can view basic information on all jobs, including jobs submitted by other users, but can only view detailed information on or modify jobs submitted by their own user IDs. Only administrators can modify jobs submitted by other users.

User commands providing information on all jobs

Any LSF user can run the following commands to view basic information on any jobs running in the cluster, including jobs submitted by other users:

bjobs
: displays information about LSF jobs. By default, `bjobs` displays information about your own pending, running, and suspended jobs. You can view information on jobs submitted by other users by using the `-u` option to specify a specific user, user group, or all users (using the `all` keyword).

bhist
: displays historical information about LSF jobs. By default, `bhist` displays historical information about your own pending, running, and suspended jobs. You can view historical information on jobs submitted by other users by using the `-u` option to specify a specific user, user group, or all users (using the `all` keyword).

bhosts
: displays information on hosts, including job state statistics and job slot limits. By default, you can view the number of jobs running on each host, including jobs submitted by other users; however, you only see the total number of jobs running on the host, not the specific users who submitted the jobs.

bqueues
: displays information on queues, including job slot statistics and job state statistics. By default, the user can view the number of jobs running in each queue, including jobs submitted by other users; however, you only see the total number of jobs running in the queue, not the specific users who submitted the jobs.

User commands that restrict information on jobs submitted by other users

Any LSF user can run the following command to provide detailed information on jobs running in the cluster, but not on jobs submitted by other users:

bpeek

displays standard output and standard error output that have been produced by unfinished jobs. This command displays detailed information on the progress of a job, but you can only view jobs that belong to your own user ID.

Queue and administrator commands that modify all jobs

Queue administrators and LSF administrators can run the following commands to modify jobs submitted by any user. LSF users can also run these commands, but only to modify their own jobs with certain restrictions:

bbot

moves a pending job relative to the last job in the queue.

btop

moves a pending job relative to the first job in the queue.

Platform LSF administrator commands that modify all jobs

LSF administrators can run the following commands to modify jobs submitted by any user. LSF users can also run these commands, but only to modify or control their own jobs with certain restrictions:

bchkpnt

checkpoints one or more checkpointable jobs. LSF administrators can checkpoint jobs submitted by any user.

bkill

sends a signal to kill unfinished jobs.

bmod

modifies job submission options of a job.

brestart

restarts checkpointed jobs.

bresume

resumes a suspended job.

bstop

suspends unfinished jobs.

Job data files

Jobs running in the LSF cluster inherit the environment from the user that submitted the job. Work files and output files are created based on the file permissions environment of the user (such as `umask` in POSIX environments). LSF does not provide additional security to these files. Therefore, to increase the security of work and output data, you need update the security of your hosts and file system according to the operating systems on your hosts.

Accessing remote hosts

By default, LSF provides commands for running tasks on remote hosts using LSF daemons (`lim` and `res`) and LSF ports (LSF_LIM_PORT and LSF_RES_PORT) for communication. Therefore, even if your cluster restricts users from directly logging into or running commands on remote hosts (therefore restricting your users to using LSF batch commands to access remote hosts), users can still run the following commands to run tasks on remote systems under certain circumstances.

lsrun runs an interactive task on a remote host through LSF. You can run a single task on a single remote host.

lsgrun runs a task on a set of remote hosts through LSF. You can run a single task on multiple remote hosts.

ch changes the host on which subsequent commands are to be executed. You can change tasks to run on a selected remote host.

False requests

LSF clusters may be vulnerable to large-scale denial of service (DOS) attacks. If one of the LSF daemons becomes overloaded with false requests, it may not be able to respond to valid requests.

By default, LSF refuses to accept client requests from hosts not listed in `lsf.cluster.`*`cluster_name`*. If LSF daemons are started on the unlisted host, the daemons will

continue to retry the connection. The LSF master host rejects these requests, but if there are many unlisted hosts doing the same thing, it may become overloaded and be unable to respond to valid requests.

Since LSF can handle large clusters (several thousand hosts in a cluster) and is designed to be resistant to this type of attack, a malicious attack needs to simulate a larger scale of false hosts in order to be successful, but LSF still remains potentially vulnerable to a very large-scale attack.

Authentication

In LSF, authentication can come by means of external authentication using the LSF `eauth` executable, or by means of identification daemons (`identd`). External authentication provides the highest level of security and is the default method of authentication in LSF. It is installed in the directory specified by the LSF_SERVERDIR parameter in the `lsf.conf` file.

By default, `eauth` uses an internal key to encrypt authentication data, but you may use a customized external key to improve security. You can also write your own `eauth` executable to meet the security requirements of your cluster, using the default `eauth` as a demonstration of the `eauth` protocol.

Kerberos integration

You can optionally configure your LSF cluster with the Kerberos version 5 integration, which provides full support for Kerberos authentication for your cluster and for MultiCluster environments. Contact Platform Computing for details.

Secure your Platform LSF cluster

Perform the following tasks to secure your LSF cluster.

Note:

If you are running LSF in a mixed cluster, you must make sure that `lsf.conf` parameters set on UNIX and Linux match any corresponding parameters in the local `lsf.conf` files on your Windows hosts. Therefore, when you need to edit the `lsf.conf` file, be sure to specify the same parameters for UNIX, Linux, and Windows hosts.

Secure communications between daemons and commands

To deal with spoofing attacks in your cluster, enable strict checking of communications between LSF daemons and between LSF commands and daemons.

You need to shut down all hosts in the LSF cluster to enable strict checking.

If you are running a MultiCluster environment, you must enable strict checking in all clusters.

1. Shut down all hosts in the LSF cluster.

   lsfshutdown
2. Edit the `lsf.conf` file.
3. Enable strict checking by specifying the LSF_STRICT_CHECKING parameter.

   Add the following line to `lsf.conf`:

   ```
   LSF_STRICT_CHECKING=Y
   ```
4. Start up all hosts in the LSF cluster.

   lsfstartup

Your LSF cluster now requires an LSF-generated checksum for all communications.

Encrypt transmission of Platform LSF commands for remote execution and login

By default, certain LSF commands use `rsh` for remote execution and `rlogin` for remote login, both of which are not encrypted. To secure these LSF commands, enable the use of `ssh` for remote execution, because `ssh` provides encryption when transmitting LSF commands.

The following LSF commands are covered by this change:

- `badmin hstartup`
- `bpeek`
- `lsadmin limstartup`

- `lsadmin resstartup`
- `lsfrestart`
- `lsfshutdown`
- `lsfstartup`
- `lslogin`
- `lsrcp`

1. Edit the `lsf.conf` file.
2. Change the remote execution shell from `rsh` to `ssh` by specifying the LSF_RSH parameter.

   For example,

   ```
   LSF_RSH="ssh -o 'PasswordAuthentication no' -o 'StrictHostKeyChecking no'"
   ```
3. Change the remote login shell by specifying the LSF_LSLOGIN_SSH parameter.

   ```
   LSF_LSLOGIN_SSH=yes
   ```
4. Reconfigure LIM and restart `mbatchd` on the master host to activate these changes.

   lsadmin reconfig

   badmin mdbrestart

The affected LSF commands now use `ssh` for remote execution and remote login.

Restrict user access to remote hosts

Even if your cluster restricts users from directly accessing remote hosts, they can still use `lsrun`, `lsgrun`, and `ch` to run tasks on specific remote hosts.

To prevent users from accessing specific remote hosts and let LSF control which remote hosts are being used, restrict access to the `lsrun`, `lsgrun`, and `ch` commands.

1. Edit the `lsf.conf` file.
2. Restrict user access to the `lsrun` and `lsgrun` commands by specifying the LSF_DISABLE_LSRUN parameter.

   ```
   LSF_DISABLE_LSRUN=Y
   ```

   LSF administrators still have access to `lsrun` and `lsgrun`.
3. Reconfigure LIM and restart `mbatchd` on the master host to activate these changes.

   lsadmin reconfig

   badmin mdbrestart
4. Restrict access to the `ch` commands by restricting the execution permissions of the `ch` binary in the LSF binary directories to the LSF administrators.

Only LSF administrators can run `lsrun` and `lsgrun` to launch tasks in remote hosts, and only LSF administrators can run `ch` to change the remote hosts on which a task runs.

Secure your cluster against false requests

To secure your cluster against false requests sent from unlisted hosts, restrict access to the LSF master host and master candidates.

The parameters you set to restrict access depend on whether your cluster allows dynamic hosts.

1. Edit the `lsf.conf` file.
2. Limit the number of master candidates in your cluster that are specified by the LSF_MASTER_LIST parameter.
3. If your cluster does not allow dynamic hosts, prevent unlisted hosts from sending requests by specifying the LSF_REJECT_NONLSFHOST parameter.

   ```
   LSF_REJECT_NONLSFHOST=yes
   ```
4. Edit the `lsf.cluster.`*`cluster_name`* file.
5. Limit or remove the range of IP addresses that are allowed to be dynamic LSF hosts by editing or deleting the LSF_HOST_ADDR_RANGE parameter.
   - If your cluster allows dynamic hosts, limit the range of IP addresses that are specified by the LSF_HOST_ADDR_RANGE parameter.
   - If your cluster does not allow dynamic hosts, ensure that the LSF_HOST_ADDR_RANGE parameter is not specified.
6. Reconfigure LIM and restart `mbatchd` on the master host to activate these changes.

   lsadmin reconfig

   badmin mdbrestart

Customize external authentication

By default, `eauth` uses an internal key to encrypt authentication data, but you may whish to use your own external key to further improve security.

You can also write your own external authentication application to meet the security requirements of your cluster.

1. Edit the `lsf.sudoers` file.
2. Use a custom external key by specifying the LSF_EAUTH_KEY parameter.

   ```
   LSF_EAUTH_KEY=key
   ```
3. Restart the cluster to activate this change.

   lsfrestart

Enable external authentication of Platform LSF daemons

You can increase LSF daemon security in your cluster by enabling LSF daemon authentication.

1. Edit the `lsf.sudoers` file.
2. Enable LSF daemon authentication by specifying the LSF_AUTH_DAEMONS parameter.

   ```
   LSF_AUTH_DAEMONS=Y
   ```
3. Reconfigure the master host to activate this change.

   badmin reconfig

Secure the cluster from root access for batch interactive jobs in pseudoterminals

Batch interactive jobs in psuedoterminals (that is, jobs submitted using `bsub -Is` and `bsub -Ip` commands) could obtain root privileges to your cluster due to environment variables (LD_PRELOAD and LD_LIBRARY_PATH) contained in the jobs.

To enhance security against users obtaining root privileges using batch interactive jobs in psuedoterminals, enable the cluster remove these environment variables from batch interactive jobs during job initialization. These environment variables are put back before the job runs.

1. Edit the `lsf.conf` file.
2. Enable the cluster to remove the LS_PRELOAD and LD_LIBRARY_PATH environment variables from jobs submitted using `bsub -Is` and `bsub -Ip` commands during job initialization by specifying the LSF_LD_SECURITY parameter.

   ```
   LSF_LD_SECURITY=y
   ```
3. Reconfigure LIM and restart `mbatchd` on the master host to activate these changes.

   lsadmin reconfig

   badmin mdbrestart

In jobs submitted using `bsub -Is` and `bsub -Ip` commands, the LD_PRELOAD and LD_LIBRARY_PATH environment variables are moved to the LSF_LD_PRELOAD and LSF_LD_LIBRARY_PATH environment variables and are moved back before the job runs.

Restrict user access to administration commands and log files

Log files may contain sensitive cluster information that need to be restricted to LSF administrators only. To restrict access to the LSF c luster log files, restrict the read/write permissions to all files in the `log` directory.

Cluster administrative tools (`badmin` and `lsadmin`) can only be used by LSF administrators. To provide an additional layer of security to prevent unauthorized administrator access to your LSF cluster, restrict the execution permissions for these commands.

1. Restrict access to the LSF cluster log files by restricting the read/write permissions of the `log` directory to the LSF administrators.
2. Restrict access to the administrative tools by restricting the execution permissions of the `badmin` and `lsadmin` binaries in the LSF binary directories to the LSF administrators.

   Tip:

   You can also restrict access to other LSF commands by restricting the execution permissions of their respective binary files.

Only LSF administrators can read the contents of the log directory or run cluster administration commands (`badmin` and `lsadmin`).