# SocketPathSecurity

## Socket Path Security

On Linux and potentially other Unix platforms, Apache Hadoop can support higher performance access to HDFS data via [[Unix domain sockets|https://en.wikipedia.org/wiki/Unix_domain_socket].

These objects live in the unix filesystem, and, when opened by the both the datanode and a local process (such as HBase), allows the local process to

1. Bypass the TCP stack for less communications overhead.
2. Share file descriptors so that read operations may actually be done in the local process.

To ensure data security and integrity, Hadoop will not use these sockets if the filesystem permissions of the domain socket are inadequate.

If you were referred to this page by an exception in the Hadoop logs, then Hadoop considers the configuration of the domain socket insecure.

This means

1. Nobody malicious can overwrite the entry with their own socket. The entire path to the socket must not contain any world-writeable directory.
2. No entry in the path is group writeable, except in the special case that the owner is root (and of course the group must be one containing

only trusted accounts)

1. The owner of the file is neither root nor the "effective user" trying to work with the socket.

All these requirements are checked, and attempts to use Domain Sockets will fail if they are unmet.

They can be addressed through tightening the permissions and changing user and group details. The exceptions should provide enough information to help you get started here.

Finally, these are not problems in the Hadoop code, they are related to the configuration of your servers. Filing bugs about these exceptions is likely to result in them being closed as InvalidJiraIssues