# **Cassandra: Security Topics to Discuss with Community**

|  |  |  |
| --- | --- | --- |
| **Security Topic/Question** | **Oleg’s Understanding** | **Community Comment** |
| Can Cassandra be configured to perform WAL (write ahead logging) in case of system failure? | No, Cassandra doesn’t support this feature |  |
| How Cassandra protects data in shared resources from unauthorized access? Is it a possibility that one user’s session would be able to access another user session’s memory, uncommitted data, etc? | Cassandra releases memory stacks used by session processes back to OS without zeroing it out. If sessions are directly associated with processes, they are protected from each other by security kernel, which would terminate offending process, trying to access another process’s memory stack. Now, if they are not, there may be a room for cross session access here. Understanding of how sessions work in Cassandra, what is the session and what kind of memory it has would help here |  |
| Is any other mechanism, except TRIGGER, Cassandra is able to provide for input validation? | No, only input validation mechanism Cassandra has is TRIGGER, linked to the table, triggering input validation event, which should correspond to Java class user would have to write on his own. |  |
| Does Cassandra support authenticators caching? | No, Cassandra doesn’t support anything like that |  |
| Cassandra maintains separate execution domain for each executing process. Is it a true statement? | Yes, it is. It is more of OS function, security kernel of OS, to be particular. Separate address space is maintained by it per each executing process, addresses are abstracted by OS memory manager from application, spawning the process. |  |
| How exactly is process of reception of transmitted information in Cassandra going? To be particular, what happens in between the moment package is received by OpenSSL and decrypted and the moment it ends up on file system? Where in the memory it exists? For how long? Who has an access? | I do not have any idea here. There is an example of answer I’m looking for (taken from my PostgreSQL assessment): “Every connection in PostgreSQL DB is associated with one process. When network package arrives via OpenSSL, data from it is loaded directly into process specific memory stack, located in RAM. In RAM data is protected by OS security kernel which would terminate any process, other than owner process, if it would try to gain an access to this data. There is no moment in between where data are not protected and vulnerable.” |  |
| How Cassandra performs protection against DoS attack? Is it a way to limit amount of concurrent connections per user? If no, what are suggested numbers for native\_transport\_max\_concurrent\_connections and native\_transport\_max\_concurrent\_connections\_per\_ip to be set to? (Suggested means, Cassandra might confidently support that amount of concurrent connections without drawing the host resources) | From what I read here: <https://issues.apache.org/jira/browse/CASSANDRA-7231> it appears the limit might be 2^15 - 1. |  |
| Can I add to logback encoder pattern following parameters: hostname, IP of host, IP of client? | From here: <http://logback.qos.ch/manual/configuration.html> it looks like I should be able to add at least hostname and IP of the host, as ${HOSTNAME} and ${SERVER\_IP}, SERVER\_IP has been predefined as `hostname -i` and exported to environment variables. I was able to get to work hostname, but not SERVER\_IP and found nothing on logging client IP. |  |
| Can I log connections and disconnections? I currently do not see them in my log file. | I’m not sure, wasn’t able to find any mention of it in logback configuration and didn’t find anything like that logged by default at my log file with level of logging turned all the way to up to “all” |  |
| Logging a lot of extra, how to get rid of it? | I was forced to set logging level to “all” since that is the only way to record SELECT and LIST queries. However, I’m not interested in whole bunch of DEBUG or TRACE messages. Is it the way to separate what I want and keep this class logger to ‘all’ and deem everything else down to ‘info’ based on the class name? If so, what class names I should look for? |  |
| Is it possible to configure Cassandra to shut down in case of audit failure? | My understanding – no. |  |
| Is it possible to configure Cassandra logging (logback) to overwrite audit logs (oldest first) in case if it is about to run out of logging space? | My understanding – no. |  |
| Is it an internal command in Cassandra (or anything external, I can make use of), which would allow to list all extension installed with DB or associated with it? | I’m looking for analogy of “\dx+” in PostgreSQL in ideal case scenario here. |  |
| Is it a way to disable integrated, but unused components, which are impossible to uninstall Cassandra DB | My understanding, core installation doesn’t have such components. |  |
| Does Cassandra support access to external executables (from inside the DB)? | I’m speaking here about some kind of EXTPROC or xp\_cmdshell analogy. |  |
| Where Cassandra stores session identifiers? Goal – check that they were invalidated after session termination. | Didn’t find much about it in Cassandra documentation. The only mention of it is TRACING, but I failed to enable it on my system, getting “10 seconds” error. |  |
| How Cassandra protects “trusted path” for session id generation? Meaning, how do we making sure only system generated session id would be taken, and no attacker generated id would be accepted? | Information on how session overall are kept, where session ids are stored, when and how they are generated and terminated, and how they are protected, once stored would help greatly here. |  |
| Mirror question to the reception one: How exactly is process of preparation for transmission of information in Cassandra going? To be particular, what happens in between the moment data is extracted from file system and encrypted by OpenSSL? Where in the memory it exists? For how long? Who has an access? | I do not have any idea here. There is an example of answer I’m looking for (taken from my PostgreSQL assessment): “Every connection in PostgreSQL DB is associated with one process. When network package is prepared for transmission via OpenSSL, data to it is loaded directly from PostgreSQL DB process specific memory stack, located in RAM. In RAM data is protected by OS security kernel which would terminate any process, other than owner process, if it would try to gain an access to this data. There is no moment in between where data are not protected and vulnerable.” |  |