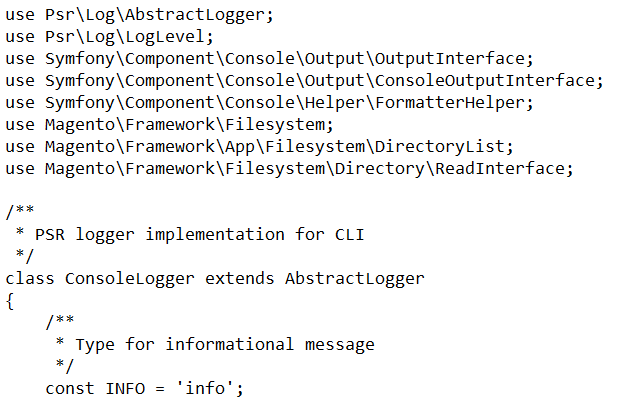
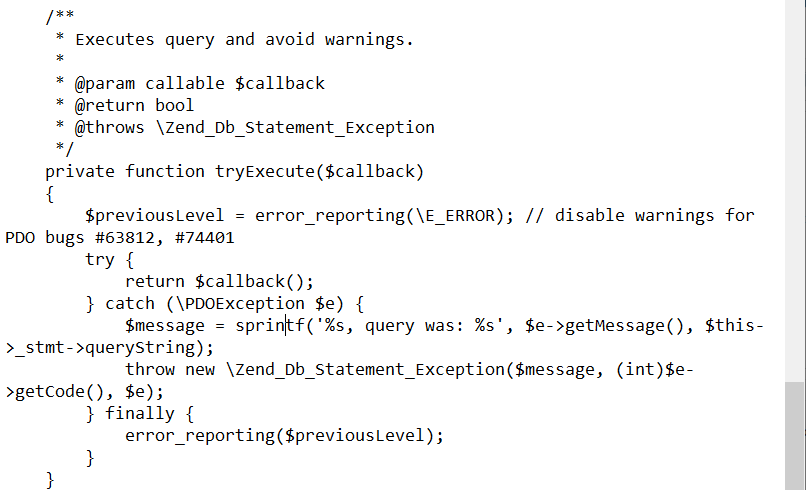
**Manual code analysis:**

**Activity Log:**

We selected some of the major CWE’s related to logs, for manual code review because from earlier misuse and assurance cases we understood that the Magento has a log capturing mechanism. Magento admin monitors and validates the logs for any unusual activities. So, we wanted to check for any weakness in the logs related source code. Configuring loggers is security-sensitive, logs are useful to track after any security incident. Capturing, maintaining proper logs is essential and it should have enough information to lead on the root cause of the damage. Codes related to logs capturing can also become targets for attackers as these may contain sensitive information. Hence configuration of what type of information and how they are logged becomes crucial.

As part of this, we shortlisted some of the logs related to CWE’s. We analyzed a few of the logs related codes to validate for any log injection attacks. We understood from the CWE- 532 that the log capturing can also become a weak point if sensitive information is exposed to the attacker. We also checked for any weaknesses mentioned in CWE-117 and CWE-778, checked for related weaknesses in some of the logs related source code such as ConsoleLogger.php, SampleDataDeployCommand.php, Mysql.php, and LoggerProxy.php.





We tried to check for any hard-coded values and how logs/errors are getting reported. Magento uses Psr\Log\LoggerInterface to capture and write logs. We didn’t observe any full path, data type conversion, or any username passwords, location information in easily decodable format in the source codes. We tried to look for those weak code suggestions from CWE and traversed the codebase; could not find any. As mentioned, since the Magento codebase is huge and our team has no hands-on experience on PHP, we were able to do only limited surf for manual code analysis.