Can you walk me through a bug that you found?

I'll walk through a sev-1 bug that we encountered that made the test infrastructure unavailable for approximately two hours.

Background:

The Avi Load Balancer Appliance has a control plane (up to 3 controller VMs) and a data plane (1 or more service engine VMs). The "WebApp" service on the controller receives RESTful API requests from the clients such as the Avi Web UI or any other user built applications (such as our test infrastructure), processes the requests, and responds back to the clients.

On the controller, the configuration is managed using the Google Protobufs.

Each of the 3 components of the Avi System, namely, the controller, service engine, and the UI have software versions associated with them. They can all be the same in which case there is no incompatibility or they can be different in which case the WebApp runs code to determine the compatibility and responds accordingly. If it is determined that the controller supports the client version, the client can subsequently set its version number and proceed to creating the configuration (such as virtual services and pools etc) using POST data. In compatible versions, the data between the client and the controller is transformed in both directions to match the Google Protobuf definitions of the client and the controller.

Bug and Analysis:

For running the feature tests, the test infrastructure first makes a get request to the controller to determine its version. Then uses this information to provision configuration. We chose this bootstrapping because this allows the test infrastructure to run tests against any controller without knowing its version ahead of time.

On one Friday morning, none of the systems including the developer environments nor the daily Jenkins jobs were able to run tests. In other words, all the tests in any testsuite would fail. Since certain sanity checks associated with the controller and service engine such as reachability have passed, no testbed creation issues were suspected.

Initially, we suspected the test infrastructure and examined its log files. From the logs, it can be seen that the get_api_version function that makes the REST API call times out leading to the infrastructure using a fallback version which is intentionally set to an incompatible version leading to subsequent incompatible REST APIs resulting in client error REST responses. Then our focus switched to why the get_api_version REST call failed. By examining the WebApp

service logs on the controller, it can be clearly seen that the API raised an exception due to a controller software bug in the WebApp component.

Lessons Learned:

Smoke Tests:

This incident prompted us to implement smoke tests that would run under 30 minutes which would be run as a first step before triggering daily jobs. In addition, these tests were run frequently on the code branch to determine the health of the software.

Commit Time Tests:

Secondly, this also prompted us to develop commit time tests to detect such bugs much earlier in the software development process.