Postmortem/root cause analysis report

Issue Summery

A Jira issue (TH-64669) was logged that the customer data got changed into the database without any proper change request. 486,000 records were affected.

Timeline

10:30 AM: The affected records were discovered

3:45 PM: The affected records were discovered

Root Cause

A deep investigation showed that an endpoint (/api/username/update) of that microservice was vulnerable to SQL injection and the database was affected due to an external attack.

Resolution & Recovery

Later, the development team fixed the bug with a quick patch, affected records were restored to the previous state from the daily backup and the issue got resolved.

Corrective & Preventive Measures

- Try to avoid putting user given input straight away to the SQL statements. Making use of parameterized query and fetch user's data indirectly in your database would protect your system from SQLi
- Try to keep all the data (confidential data) in an encrypted format in your database. In case attackers gain access to your backend system through SQLi, they will not be able to exfiltrate sensitive set of information.
- You can smartly code your backend to escape out or filter out those characters which need to be escaped.
- Access to the database should be at a bare minimum level as per requirement. Also, share the backend code to the least possible members of your technical team.

Prepared by Nazibur Rahman

pythonboy007@gmail.com