ERROR 500: It's a Crisis

Issue Summary:

- Duration: The outage occurred *from 10:00 AM to 11:30 AM (UTC-5)*.
- Impact: The primary web service was completely down during this period, affecting all users. Approximately 100% of users were affected.
- Root Cause: The root cause of the outage was a human mistake where an incorrect file extension was used in a critical configuration file.

Timeline:

10:00 AM: The issue was detected when users started reporting that they were unable to access our web service.

10:05 AM: Engineers noticed a spike in error logs but initially attributed it to normal fluctuations.

10:15 AM: As user complaints continued to pour in, the engineering team began investigating the issue. They first suspected a potential server overload.

10:30 AM: After investigating server resources and database performance, the team couldn't find any anomalies, but the problem persisted.

10:45 AM: A senior engineer joined the investigation and discovered a misconfiguration in the server settings file.

11:00 AM: The incident was escalated to the DevOps and System Administration teams to assist with a more detailed examination.

11:15 AM: The misconfigured file was corrected, and the service was brought back online.

11:30 AM: The service was fully restored, and users reported that they could access it without any issues.

Root Cause and Resolution: The root cause of the outage was a file extension error in a critical configuration file. A junior developer mistakenly saved the configuration file with the **".conf"** extension

instead of ".config". As a result, the application couldn't read the configuration settings, leading to a complete service outage.

To resolve the issue, the following steps were taken:

- The misconfigured file was identified and corrected, with the proper ".config" extension applied.
- The web server was restarted to apply the corrected configuration.
- The service was monitored for a brief period to ensure that it was operating as expected.

Corrective and Preventative Measures: To prevent similar incidents in the future, the following measures will be implemented:

- Code Review and Quality Control: All configuration files and code changes will undergo thorough code reviews to catch errors like file extensions early in the development process.
- File Extension Policy: Establish a clear file extension naming convention and documentation to ensure consistency across the team.
- Automated Monitoring: Implement monitoring and alerting systems to immediately detect service outages or abnormal behavior.
- Rollback Plan: Develop a rollback plan to quickly revert to the previous working configuration in case of errors during deployments or updates.

Tasks to Address the Issue:

- Conduct a post-incident review meeting to analyze the entire incident, identify areas of improvement, and assign action items.
- Develop and document a file extension policy to prevent similar errors in the future.
- Implement automated monitoring and alerting to promptly detect and respond to service issues.
- Establish a rollback plan for configurations and updates.
- Provide training and guidance to team members, emphasizing the importance of careful file handling and naming conventions.

By implementing these corrective measures and proactive steps, we aim to minimize the likelihood of human errors causing service outages and improve our incident response capabilities.

Amine Abahmane .