**Analyzing the Software Bug Incident and Proposing Preventive Measures**

**Eshaan Singh**

**Washington State University**

**Introduction**

On [specific date], a bad software update from CrowdStrike, a well-known cybersecurity company, caused problems in many industries, especially the aviation sector. This situation shows why it’s so important to test software properly before releasing it, especially when it’s used in critical areas like transportation. CrowdStrike is famous for its tools that protect computers and detect cyber threats, but this time, an issue with their software interacting with Microsoft’s systems created a serious problem.

Microsoft, whose operating systems are used by many businesses, works with companies like CrowdStrike to make sure everything works smoothly together. In this case, the connection between CrowdStrike’s update and Microsoft’s systems didn’t go well, leading to big challenges for companies relying on both.

**Incident Analysis**

The problem happened because of an update CrowdStrike released that didn’t work well with Windows operating systems. This caused systems to crash or stop working properly. In the aviation industry, this bug disrupted important communication systems, which are critical for daily operations. Airlines that depended on these systems experienced major delays and issues with scheduling.

For example, at some airports, flight schedules and communication tools stopped working, creating confusion and delays. Ground control systems, which need real-time updates, were also affected. This showed how important it is for software used in critical industries to be reliable.

**Preventive Measures**

1. **Better Testing Before Releases:**  
   Companies should test updates in real-world-like situations before rolling them out. This includes trying them on all major systems used in important areas like aviation. Crowdsourcing testing could also help because it would involve a wider range of users finding issues early.
2. **Improved Communication Between Companies:**  
   CrowdStrike and Microsoft, or any other companies working together, need to talk more during development. They should check for any possible issues and make plans to fix them quickly if they happen.
3. **Automatic Backup Plans:**Updates should include tools to automatically roll back to the last working version if something goes wrong. This would limit the damage and keep systems running.
4. **Using AI to Find Bugs:**Artificial intelligence could be used to check for potential problems in updates before they go live. AI can run through tons of scenarios to find errors that people might miss.
5. **Quick and Clear Communication in Emergencies:**If a problem does happen, companies need to share information quickly with everyone involved. They should explain what happened, suggest temporary fixes, and provide customer support to help affected users.

**Conclusion**

The CrowdStrike update problem showed how small mistakes in software updates can have big impacts. The aviation industry’s struggles during this incident highlight why software testing and communication are so important. The key lessons are to improve testing, make sure companies work together better, and create systems that can roll back bad updates.

This situation should remind everyone how important it is to invest in quality testing and prepare for problems before they happen. With better preparation and teamwork, we can avoid problems like this in the future and keep important systems running smoothly.

**References:**

1. Jones, D. (2024, July 19). *CrowdStrike software update at the root of a massive global IT outage*. Cybersecurity Dive. <https://www.cybersecuritydive.com/news/crowdstrike-microsoft-global-IT-outage/721874>
2. Shepardson, D. (2024, October 25). *Delta sues CrowdStrike over software update that prompted mass flight disruptions*. Reuters. <https://www.reuters.com/legal/delta-sues-crowdstrike-over-software-update-that-prompted-mass-flight-2024-10-25>
3. *CrowdStrike executive apologizes to Congress for July global tech outage*. (2024, September 25). AP News. <https://apnews.com/article/aa1e9c84ee34bc38aca69731d9d3b9a7>
4. Sider, A., Tangel, A., & McMillan, R. (2024, September 3). *The Day Delta’s “On-Time Machine” Broke, and the Blame Game It Sparked*. WSJ; The Wall Street Journal. <https://www.wsj.com/business/airlines/the-day-deltas-on-time-machine-broke-and-the-blame-game-it-sparked-b462fc80>