**FORMAL METHOD IN SOFTWARE  
 ENGINEERING**



**ASSIGNMENT # 01**

*Submitted By*

Hifza Majeed

(19P-1652)

*Submitted to*

MR. SHAKIR ULLAH SHSH

**DEPARTMENT OF COMPUTER SCIENCE**

**FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCINCES, PESHAWAE**

**Airline Pilots Are Making Errors Back in the Air**

Covid has meant months out of the cockpit. As countries begin to open up, mid-air mistakes are  
mounting.

**Back in the cockpit after time off recovering from Covid-19,** an airline pilot [forgot to start](https://asrs.arc.nasa.gov/search/database.html) his plane’s second engine for takeoff, a mistake that could have ended in disaster if he hadn’t aborted the flight.

**The incidents are among dozens of mistakes**, [confidentially declared](https://asrs.arc.nasa.gov/index.html) by out-of-practice pilots since the start of the pandemic, **that are stored on a low-profile database designed to identify emerging safety threats**. The [monitoring program](https://asrs.arc.nasa.gov/index.html), funded by the Federal Aviation Administration, is decades old but is now flashing warning signs as planes return to the skies across the world.

**Heartbleed Bug**

Appearing in the OpenSSL library, the Heartbleed Bug is a dangerous security vulnerability. The Transport Layer Security (TLS) protocol employs the OpenSSL cryptography library. Because of its widespread use in TLS, Heartbleed spread quickly. This bug allows virtually anyone on the internet to read memory on machines running affected iterations of OpenSSL. Up to 64 kb of system memory could be read. While the Heartbleed Bug was revealed to the public in **2014**, it rolled out in **2012**. Improper input validation on account of a missing bounds checks within the TLS heartbeat extension caused the bug. Since it was a bug in the heartbeat extension, the name Heartbleed thus spawned. A **2014** article in The Register reported that 1.5% of the most popular TLS-enabled sites remained vulnerable to the Heartbleed bug. However, TLS implementations aside from OpenSSL were untouched. Therefore, the Windows version of TLS and Mozilla's Network Security Services were unaffected by the Heartbleed Bug. A patch eventually fixed the problem with OpenSSL version 1.0.1g. By adding bounds checks to prevent buffer over-read, the Heartbleed Bug was successfully patched. Why it's one of the worst programming mistakes: The Heartbleed Bug created a major security threat. The time between launch and patching left affected systems vulnerable for years. Any time there's a computer vulnerability problem, this creates a huge data security concern.