Case Study 1: The Therac-25 Software Failure

1. Background of the Security Issues

Therac-25, a radiation therapy machine, experienced multiple accidents in the 1980s caused by software malfunctions. Released in 1983, it was created to provide intense radiation for treating cancer.

In contrast to the Therac-6 and Therac-20, which utilized hardware safety interlocks, the Therac-25 mainly depended on software-based safety checks. Due to the absence of adequate hardware safety checks and the presence of significant software flaws, patients suffered from massive radiation overdoses, causing injuries and fatalities.

A critical issue in the software was a race condition that caused unintended radiation doses due to rapid switching between treatment modes by operators. From 1985 to 1987, there were a minimum of six incidents that led to patients experiencing significant radiation overdoses.

2. Consequences

The consequences were severe. Patients were exposed to doses of radiation up to 100 times greater than prescribed, causing burns, paralysis, and in some cases, death. For example, a 61-year-old woman in Georgia received 20,000 rad (instead of the intended 200 rad) while being treated for breast cancer. She experienced paralysis and had to undergo breast amputation due to the overdose. Another patient in Texas experienced similar doses and died within five months of radiation poisoning.

Regulatory Enhancements: In response to these events, the FDA enforced more stringent guidelines and regulations for the creation and authorization of medical devices, particularly those that heavily utilize software. The updated rules demanded stricter testing for both the hardware and software parts of medical devices, with a focus on ensuring that the software functions properly in various scenarios and aligns seamlessly with hardware safety features.

3. Lessons Learned

Avoid depending completely on software for important safety functions.

A significant flaw in the Therac-25 design was its heavy dependence on software for regulating radiation output, without sufficient hardware interlocks. Due to complications like race conditions, memory corruption, and user errors, software is naturally susceptible to bugs and unpredictable behavior.

Hardware Interlocks: In radiation equipment, hardware interlocks limit the amount of dosage that can be administered to prevent overexposure.

Programming languages that are crucial for safety: Ada and MISRA C are employed to create more reliable and anticipated code for systems where failure is unacceptable.

Thorough Testing and Verification: AECL failed to make comprehensive testing of the software a priority, leading to the discovery of significant bugs. Not doing this results in unintended outcomes, like what happened with Therac-25's race conditions that led to fatal overdoses.

Tools such as SonarQube, Coverity, and CodeSonar permit developers to analyze codebases for bugs, security flaws, and inefficiencies.

Human-Centered Design: Platforms such as UsabilityHub and Axure assist designers in testing interfaces to guarantee that users can easily comprehend and react to errors.

Redundant Systems: A mission-critical environment includes several layers of redundancy to avoid failure from a single point of error.

Incident Reporting Systems: Numerous sectors are now utilizing automated systems such as SafetyNet or PRISMA Incident Reporter to monitor and report incidents. This enables manufacturers, regulators, and operators to tackle problems at an early stage.

Summary: The Therac-25 incident highlights the risks of relying solely on software. Modern safety-critical systems need to combine software and hardware protections, go through extensive testing and validation.

Citations:

1. Russell Taylor, "The Therac-25: A Case Study in Safety Failure" (2022)(CIIS therac).
2. "Therac-25: Software that Killed," Cambridge University(Cambridge University Press & Assessment).
3. "Therac-25 Accidents," Wikipedia(Wikipedia).
4. “The Therac-25 Accidents | A short documentary | Fascinating horror” 2023 (Youtube)