**ARKANSAS PANDEMIC UNEMPLOYMENT ASSISTANCE BREACH**

**Identify Root Cause**

Based on the recent report of a data breach incident that affected the Arkansas State Pandemic Unemployment Assistance system website, the problem was tipped off by a tech-savvy user whom found that the website has left applicants’ bank accounts, DOB, and Social Security number exposed.

According to Protech Solutions of Little Rock, the private company vendor whom the government had worked with in the past, the company had to bring in outside help in order to keep up with the existing as well as the additional work loads. Thus, resulted in the short fall of the final product due to the inconsistency in the secure software development between Protech Solutions and their outside contractor.

**Understanding SSD (Secure Software Development)**

SSD or Secure Software Development is a range of practices, tasks, and implementation that every personnel involve in the software development must follow to satisfy both external and internal security requirements of the software development cycle.

Implementation of a solid and consistent SSDF or Secure Software Development Framework across the board would provide Protech Solutions a standardized workflow that ensure every aspect of the SDLC is complying with the security requirements set by the organization and the government.

**Recommended Practices**

Based on the situation of Protech Solutions, it’s recommended that they follow these two practices to establish a seamless workflow for every personnel that involve in the development of the product.

**1st Practice: Define Security Requirements for Software Development**

The first step is to identify different security requirement from internal sources (e.g., the organization’s policies, business objectives, and risk management strategy) and external sources (e.g., applicable laws and regulations). Once these information are collected, it need to be made available and shared to every personnel involve in the development. This will ensure that every requirement is taken into account throughout the SDLC, as well as minimizing duplication of effort.

Recommended task:

* Identify all applicable security requirements for the organization’s general software development and maintain the requirements over time.

Implementation Examples:

* Define policies that specify the security requirements for the organization’s software to meet, including secure coding practices for developers to follow.
* Ensure that policies cover the entire software life cycle, including notifying users of the impending end of software support and the date of software end-of life.

**2nd Practice: Implement Roles and Responsibilities**

This step requires that every personnel that involve in the SDLC is prepared to perform their SSDF-related roles and responsibilities throughout the SDLC.

Recommended tasks:

* Create new roles and alter responsibilities for existing roles to encompass all parts of the SSDF. Periodically review the defined roles and responsibilities and update them as needed.
* Provide role-specific training for all personnel with responsibilities that contribute to secure development. Periodically review role specific training and update it as needed
* Obtain upper management commitment to secure development and convey that commitment to all with SSDF-related roles and responsibilities.

Implementation Examples:

* Define SSDF-related roles and responsibilities for all members of the software development team.
* Define roles and responsibilities for cybersecurity staff, security champions, project managers and leads, senior management, software developers, software testers/quality assurance personnel, product owners, and others involved in the SDLC.
* Conduct an annual review of all roles and responsibilities.
* Document the desired outcomes of training for each role.
* Create a training plan for each role.
* Assist upper management in incorporating secure development support into their communications with personnel with SSDF-related roles and responsibilities.