**Case Study Analysis: Capital One Data Breach**

**By: Darrell Walker**

**Introduction**

**Name of case and link:** Capital One Data Breach (2019)  
**Link:** https://www.cbsnews.com/news/capital-one-data-breach-what-you-need-to-know/  
**Date of case:** July 2019

**Why did this case make the news?**  
This case made the news because it affected over 100 million people in the U.S. and Canada. A hacker stole private information like names, addresses, credit scores, and Social Security numbers from people who applied for credit cards with Capital One. Since Capital One is a big bank that handles sensitive data, this was a huge deal and got a lot of media attention.

**Describe the breach**  
The breach happened because of a problem with Capital One’s cloud server settings. The hacker used a method called “server-side request forgery” to trick the system and get access. Capital One was using Amazon Web Services (AWS) to store their data, and their firewall was not set up correctly. This allowed the hacker to reach private data folders that should have been protected.

**Why was this company a target?**  
Capital One was a target because it had a lot of valuable personal data from millions of people. The hacker knew there was a weakness in how the company had set up its cloud security and used that to break in. Since banks store financial and identity information, they are common targets for cyberattacks.

**Identify the threats**

* **Immediate threats:** The hacker took personal data, which could be used for identity theft, fraud, or scams.
* **Potential threats if the vulnerability goes unresolved:** If Capital One hadn’t found the problem, the hacker might have stayed in the system and stolen even more data. Other companies using the same setup might also be at risk if they don’t fix similar security issues.

**What could a developer have done to prevent this breach?**  
The developers could have better protected the firewall and reviewed the security settings more carefully. They should have used tools to detect strange behavior in the system, like someone trying to access hidden files. Regular testing and updates would have helped fix this problem before the attack happened.

**Which policy or policies will help prevent this type of attack?**

* **Access control policies** to make sure only the right people can view or change data.
* **Patch management** to fix known security problems quickly.
* **Governance policies** to guide teams on how to protect data and do regular system checks.

**Summarize the case by explaining the role of best practices, Triple A and defense in depth in preventing future attacks.**  
This case proves why following security best practices is important. Developers should use the **Triple A model**:

* **Authentication** checks if someone is really who they say they are.
* **Authorization** controls what users are allowed to do in the system.
* **Accounting** keeps track of everything users do.

Also, using **defense in depth**, which means adding many layers of security, helps stop attackers. Even if one layer fails, others can still protect the system. If Capital One had more layers in place, this hack might have been stopped before it caused so much damage.