Analyze a real-life case (e.g., a breach due to lack of HTTPS) and write 1–2 paragraphs explaining what went wrong and what protocols/security measures could have helped.

- **Case Study: Equifax Data Breach (2017)**

One of the most infamous security failures due to the lack of proper encryption was the **2017 Equifax breach**, where attackers exploited a vulnerability in the Apache Struts web framework to access sensitive personal data of **147 million people**. While the initial intrusion wasn't solely due to missing HTTPS, poor encryption practices exacerbated the breach. Sensitive data, including Social Security numbers and credit details, was transmitted or stored insecurely, allowing attackers to intercept and exfiltrate information easily.

What Went Wrong & How to Prevent It:

1. **Missing Encryption (HTTPS/TLS)**: While Equifax used HTTPS externally, internal systems lacked proper encryption, enabling attackers to move laterally. **Mandating HTTPS across all systems**, including internal communications, would have reduced exposure.
2. **Poor Patch Management**: The unpatched Apache Struts flaw allowed the breach. **Regular vulnerability scanning** and **automated patch deployment** could have prevented exploitation.
3. **Inadequate Network Segmentation**: Sensitive data wasn't isolated. **Zero Trust Architecture (ZTA)** and **strong segmentation** would have limited attacker movement.
4. **Weak Data Encryption at Rest**: Stored data wasn't fully encrypted. **AES-256 encryption** for databases could have rendered stolen data unusable.

Key Takeaway: A layered defense—HTTPS everywhere, timely patches, strict access controls, and encryption—could have mitigated this breach.

Key Security Measures That Could Have Stopped the Breach

Failure	Solution	Protocol/Tool
Unpatched Struts	Automated patch management	**WSUS, Qualys, Tenable**
No HTTPS internally	Enforce TLS 1.3 everywhere	**Let's Encrypt, HSTS**
Plaintext credentials	Secrets management	**Hashicorp Vault, AWS KMS**
No network segmentation	Microsegmentation	**Cisco ACI, Zero Trust (BeyondCorp)**
Unencrypted DB storage	TDE (Transparent Data Encryption)	**SQL Server TDE, PostgreSQL pgcrypto**