Incident Postmortem: Spring4Shell Exploit Attack

## **Summary**

* **Incident Title:** Spring4Shell Exploit Attack on NBN Connection Service
* **Timestamp:** Attack began at **12:00 PM** and was mitigated by **2:00 PM**
* **Teams Involved:** Security Operations Center (SOC), Networks Team
* **Severity:** **High** – Service downtime and security breach risk

## **Impact**

* The **NBN Connection (nbn.external.network)** service was compromised due to a **remote code execution (RCE) attack**.
* Service **downtime for 2 hours**, affecting critical infrastructure.
* Potential risk of **data leakage and system takeover** if left unmitigated.

## **Detection**

* Detected via **firewall logs**, which showed **suspicious POST requests to** /tomcatwar.jsp.
* The **Security Operations Team** identified malicious payloads exploiting **Spring4Shell (CVE-2022-22965)**.

## **Root Cause**

* The attack leveraged the **Spring4Shell vulnerability**, allowing attackers to upload a malicious JSP web shell.
* Lack of **WAF (Web Application Firewall) rules** to filter such requests beforehand.
* Exploitable **misconfiguration in the affected service**, enabling attackers to bypass security measures.

## **Resolution**

* Implemented **firewall rules** to block:
* POST requests to /tomcatwar.jsp
* Requests containing class.module.classLoader.resources.context.parent.pipeline.firstPattern
* Suspicious User-Agent headers linked to automated bots
* Conducted **security patching and server hardening** to prevent further exploitation.

## **Action Items**

**Short-Term:**

* Continue **monitoring logs** for any further attack attempts.
* Conduct **security audit** on all Spring-based services to ensure no other instances are vulnerable.

**Long-Term:**

* Apply **Spring Framework patches** and update affected systems.
* Deploy a **Web Application Firewall (WAF)** with **deep packet inspection** to filter similar attacks.
* Implement **continuous security training** for developers to follow secure coding practices.