## Incident Postmortem: Spring4Shell Malware Attack on NBN Connection Service

## **Summary**

On **March 20th, 2024, at 14:20 UTC**, Telstra’s Security Operations Centre (SOC) detected a **P1 - Critical** malware attack targeting the **NBN Connection service (nbn.external.network)**, which operates on **Spring Framework 5.3.0**. The attack involved multiple malicious POST requests to the /tomcatwar.jsp endpoint, exploiting the **Spring4Shell (CVE-2022-22965)** vulnerability. The incident was identified through firewall log analysis and was successfully mitigated **two hours** after detection by implementing a specific firewall rule. Key teams involved in the response included the **Security Operations Centre** and the **NBN Team**.

## **Impact**

• **Service Disruption:** The NBN Connection service experienced significant downtime, impairing high-speed internet connectivity for customers relying on this infrastructure.

• **Operational Impairment:** Critical services dependent on the NBN Connection, such as remote communications and business operations, were temporarily affected.

• **Potential Data Exposure:** Although no data breaches were confirmed, the nature of the attack posed a risk of unauthorized command execution and potential data exfiltration.

## **Detection**

The incident was discovered through routine monitoring of firewall logs by the SOC. Analysis revealed a pattern of multiple POST requests to the /tomcatwar.jsp endpoint originating from several IP addresses within the AU region. These requests contained specific malicious payloads characteristic of the Spring4Shell vulnerability, including parameters like class.module.classLoader.resources.context.parent.pipeline.first.pattern and others designed to execute remote commands.

## **Root Cause**

The root cause of the incident was the exploitation of the **Spring4Shell (CVE-2022-22965)** vulnerability within the **Spring Framework 5.3.0** used by the **NBN Connection service**. Attackers crafted malicious POST requests to the /tomcatwar.jsp endpoint, embedding payloads that leveraged this vulnerability to execute arbitrary commands on the server, leading to service disruption and impaired functionality.

## **Resolution**

To mitigate the attack, the SOC collaborated with the Networks Team to implement a targeted firewall rule using a Python-based HTTP server (firewall\_server.py). The rule specifically blocked incoming POST requests to the /tomcatwar.jsp endpoint that contained the identified malicious parameters. This measure effectively halted the ongoing attack within two hours of its initiation, restoring the NBN Connection service to operational status and preventing further unauthorized access.

## **Action Items**

1. **Immediate Actions:**

• **Firewall Rule Implementation:** Successfully deployed a Python-scripted firewall rule to block malicious POST requests targeting the /tomcatwar.jsp endpoint.

• **Service Restoration:** Coordinated with the Networks Team to ensure the NBN Connection service was promptly restored to normal operations.

2. **Short-Term Actions:**

• **Vulnerability Patching:** Upgrade the Spring Framework to the latest version to eliminate the exploited Spring4Shell vulnerability.

• **Enhanced Monitoring:** Increase the frequency and depth of firewall log reviews to detect similar or new attack patterns more swiftly.

• **Incident Documentation:** Complete detailed documentation of the incident for future reference and compliance purposes.

3. **Long-Term Actions:**

• **Security Training:** Conduct training sessions for the SOC and relevant teams on identifying and responding to similar vulnerabilities and attack vectors.

• **Comprehensive Security Audit:** Perform a thorough security audit of all critical services to identify and remediate potential vulnerabilities.

• **Automation of Response Mechanisms:** Develop automated scripts and tools to detect and mitigate such attacks in real-time, reducing response times.

• **Collaboration with Development Teams:** Work closely with development teams to ensure secure coding practices are followed, particularly when using frameworks like Spring.

4. **Future Prevention:**

• **Regular Updates and Patching:** Establish a routine schedule for updating and patching all software frameworks and dependencies to minimize vulnerability exposure.

• **Advanced Threat Detection Systems:** Invest in more sophisticated threat detection and prevention systems that can identify and block complex attack patterns.

• **Incident Response Drills:** Conduct regular incident response drills to ensure all teams are prepared to handle similar attacks efficiently.

By documenting this incident, Telstra aims to strengthen its security posture, ensure compliance with governance and regulatory standards, and educate its teams to better handle future cybersecurity threats.

**Prepared by:**

Telstra Security Operations

**Date:** April 27, 2024