**Incident Postmortem: Spring4Shell Exploit Attempt on NBN Infrastructure**

**Summary**

On **March 20, 2022, starting at 03:21:00 UTC**, Telstra Security Operations identified an ongoing, critical malware attack targeting the NBN Connection infrastructure, which provides high-speed internet services to customers across Australia. The attacker exploited the Spring4Shell vulnerability in Spring Framework version 5.3.0 by sending crafted POST requests to deploy a malicious JSP web shell. The attack involved a distributed set of IP addresses ranging from attacker.ip.address.network1 to network499, making simple IP blocking ineffective. The incident engaged multiple teams, including Security Operations, NBN Infrastructure, and Networks teams. Due to the critical nature of the infrastructure affected, the incident was classified as **critical severity**. The attack was successfully contained approximately 2 hours after initial detection.

**Impact**

Although no confirmed data breach or system compromise was recorded, the attack posed a severe risk to the integrity and availability of the NBN Connection service. If successful, the attacker could have gained remote code execution privileges on the affected servers, potentially leading to service disruption, data theft, or further lateral movement within the network. The distributed nature of the attack increased the difficulty of immediate mitigation, raising the risk level until appropriate firewall rules were deployed.

**Detection**

The incident was discovered by continuous monitoring of firewall logs, which revealed repeated POST requests targeting /tomcatwar.jsp on nbn.external.network. These requests contained suspicious headers such as c1=Runtime, c2=<%, and suffix=%>// alongside payload parameters exploiting Spring’s class loader properties. The logs showed that these requests successfully bypassed firewall controls initially, signaling a sophisticated attempt to exploit the known Spring4Shell vulnerability. Rapid analysis by the Security Operations team confirmed the malicious nature of the traffic.

**Root Cause**

The root cause of this incident was the exploitation of the **Spring4Shell vulnerability (CVE-2022-22965)** affecting Spring Framework 5.3.0. The vulnerability allowed attackers to execute arbitrary code remotely by injecting malicious Java code via class loader pipeline properties in specially crafted POST requests. The affected infrastructure had not yet been patched against this vulnerability, which was publicly disclosed shortly before the attack. The lack of initial firewall rules to detect and block these specific malicious request patterns allowed the attacker to attempt web shell deployment.

**Resolution**

Upon identification, the Security Operations team immediately notified the NBN infrastructure team to initiate incident response and system assessments. Simultaneously, the Networks team was alerted and provided with detailed indicators of compromise, including header patterns and payload signatures observed in the attack. A Python-based HTTP server script simulating firewall behavior was developed and deployed, implementing precise rules to block all POST requests containing:

* .jsp paths
* Malicious headers like c1=Runtime, c2=<%, and suffix=%>//
* Payload content referencing getRuntime().exec and request.getParameter("cmd")

This proactive mitigation effectively stopped the attack within 2 hours of the initial incident. Continuous monitoring ensured no further successful exploit attempts occurred after firewall rule deployment. All teams coordinated closely to verify containment and prepare for post-incident analysis.

**Action Items**

* **Patch Management:** Immediate prioritization of patching Spring Framework on all affected systems to the latest secure version to prevent future exploitation.
* **Firewall Rule Enhancement:** Networks team to expand and standardize firewall rules across all critical infrastructure, incorporating the newly identified attack patterns.
* **Threat Intelligence Integration:** Update security monitoring tools and threat intelligence feeds with Spring4Shell signatures and indicators for faster detection in the future.
* **Incident Response Training:** Conduct knowledge-sharing sessions across SOC, Infrastructure, and Networks teams to review incident handling, attack characteristics, and mitigation strategies.
* **Continuous Monitoring:** Implement enhanced logging and anomaly detection for suspicious web shell behaviors and class loader exploitation attempts.
* **Post-Incident Review:** Schedule follow-up meetings to audit patch status, firewall rules effectiveness, and overall readiness for similar threats.
* **Documentation:** Maintain detailed incident documentation and playbooks to improve response times and communication for future incidents.