**Incident Postmortem: Spring4Shell Malware Attack on NBN Services**

**Summary:**

This postmortem report outlines the malware attack exploiting the **Spring4Shell vulnerability (CVE-2022-22965)**, which targeted externally exposed Spring Framework applications used by NBN infrastructure. The attack commenced on **2022-03-20 at 03:16:34 UTC** and was mitigated approximately two hours later after collaboration between **Telstra Security Operations**, the **nbn team**, and the **Networks team**. The incident is classified as **Severity 1 – Critical**, due to its potential for system compromise, data exposure, and service disruption.

* **Incident Start Time:** 2022-03-20T03:16:34Z
* **Incident End Time:** 2022-03-20T05:16:34Z
* **Detection Time:** 2022-03-20T03:16:34Z
* **Root Cause Fixed Time:** 2022-03-20T05:16:34Z
* **Status:** Resolved
* **Participants:** Telstra Security Operations, NBN Team, Networks Team
* **Impact Severity:** High / Critical

**Impact**

The malware attack resulted in **impaired functionality and service downtime** on the NBN infrastructure. The attacker successfully executed **remote code execution (RCE)** on the externally accessible NBN services by leveraging a zero-day vulnerability in the Spring Framework.

Potential impacts included:

* **Data Breach:** Unauthorized access to sensitive customer or system data.
* **System Takeover:** Full server control through arbitrary code execution.
* **Service Disruption:** Degraded or completely unavailable NBN services for customers.

**Detection**

The incident was detected through a combination of:

* **Firewall logs**, which indicated a sharp increase in abnormal and suspicious requests targeting the endpoint /tomcatwar.jsp using the POST method.
* **Customer complaints** regarding degraded service availability.
* **Forensic analysis**, which confirmed that remote code execution had occurred.

**Root Cause**

The root cause was the exploitation of **Spring4Shell (CVE-2022-22965)** — a newly disclosed zero-day vulnerability in the Spring Framework. At 03:16:34 UTC, the attacker began sending malicious HTTP POST requests to the endpoint /tomcatwar.jsp on the exposed address nbn.external.network. These requests included specially crafted headers and payloads allowing arbitrary code execution on affected Spring-based systems.

**Resolution**

The mitigation timeline is as follows:

* **03:16 – 03:46 UTC:** Telstra Security Operations triaged firewall alerts and confirmed exploit patterns consistent with Spring4Shell.
* **03:46 – 04:16 UTC:** Security Operations analyzed traffic and identified specific exploit signatures, which were communicated to the Networks Team.
* **04:16 – 05:16 UTC:** Networks Team developed and deployed a Python-based firewall rule that blocked requests matching the malicious pattern.

The firewall rule was configured to block:

* **Method:** POST
* **Path:** /tomcatwar.jsp
* **Headers:**
  + suffix: %>//
  + c1: Runtime
  + c2: <%
  + DNT: 1
  + Content-Type: application/x-www-form-urlencoded (based on feasibility without affecting legitimate traffic)

After the firewall was updated, a significant reduction in malicious traffic was observed, and service stability was restored. A full forensic investigation was launched immediately following mitigation.

**Action Items**

**Immediate Actions**

* Fully deploy firewall rules to block Spring4Shell exploit payloads.
* Investigate all affected systems for signs of compromise or lateral movement.
* Update all Spring-based applications to patched versions that address CVE-2022-22965.

**Short-Term Improvements**

* Review firewall rules for broader detection and mitigation of similar RCE attack patterns.
* Conduct internal training sessions for developers and system engineers on secure coding practices and vulnerability awareness.
* Improve alert correlation between firewall logs and service health monitors.

**Long-Term Strategy**

* Implement continuous monitoring solutions for anomaly detection on web traffic and server behavior.
* Regularly update the organization’s incident response playbook to incorporate lessons learned from this attack.
* Consider deploying a Web Application Firewall (WAF) for layered protection against application-layer attacks.
* Establish regular penetration testing and threat simulation exercises focused on zero-day and framework-based vulnerabilities.