**1. Executive Summary**

On March 20, 2022 at 14:16:34 AEDT, Telstra’s NBN infrastructure was hit by a distributed RCE exploit (Spring4Shell, CVE-2022-22965) via malicious POSTs to /tomcatwar.jsp. Within 2 h 15 m the attack was contained by a custom firewall rule blocking the exploit signature. Service disruption affected high-speed NBN customers, generating elevated support tickets. Key lessons include strengthening WAF rules, automating third-party patching, and enhancing threat monitoring.

**2. Incident Timeline**

|  |  |
| --- | --- |
| Time (AEDT) | Event |
| 14:16:34 | Firewall alert: high volume of POSTs to /tomcatwar.jsp |
| 14:30 | NBN Team notified; initial forensic review begins |
| 15:15 | Attack pattern identified (499 attempts; 80+ IPs) |
| 16:15 | Networks Team deploys firewall rule blocking exploit header and body signature |
| 16:31 | Attack traffic drops to zero; service functionality restored |
| 16:31–16:59 | Post-mitigation forensic scans and webshell removal confirmed |
| 16:31–16:59 | Customers serviced; support tickets reduced to baseline |

**3. Impact Metrics**

* **Total Attempts:** 499 POST requests
* **Unique Attacker IPs:** 82 in AU, 4 in US
* **Peak Rate:** 124 requests/min at 14:17
* **Customer Impact:** ~2% of active sessions degraded; average downtime 15 min
* **Detection to Root-Cause Fix:** 2 h 15 m

**4. Root-Cause Analysis**

* **Vulnerability:** Spring4Shell (CVE-2022-22965) allows data-binding on JDK 9+
* **Exploit Signature:**
  + Headers: suffix=%>//, c1=Runtime, c2=<%, DNT=1
  + Body parameter:  
    class.module.classLoader.resources.context.parent.pipeline.first.pattern
* **Patch Gap:** Framework version 5.3.0 was unpatched due to delayed test cycles
* **Supply-chain Process:** No automated dependency checks for minor releases

**5. Containment & Eradication**

* **Firewall Rule:** Block POST /tomcatwar.jsp where exploit signature appears in headers or body
* **Verification:**
  + File-integrity scans confirmed removal of tomcatwar.jsp
  + Endpoint EDR showed no ongoing arbitrary command executions
* **Post-Mitigation Scans:** No residual webshells or backdoors detected

**6. Lessons Learned**

1. **Automate Patching:** Integrate SBOM-based alerts for critical framework releases
2. **WAF Tuning:** Develop application-layer rules for payload patterns, not just IP blocklists
3. **Threat Intelligence:** Enrich firewall logs with GeoIP and exploit signature detection
4. **Cross-Team Drills:** Run quarterly tabletop exercises covering zero-day scenarios

**7. Action Items**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Action | Owner | Deadline | Status |
| 1 | Deploy signature-based WAF rules for Spring4Shell and similar exploits | Networks Team | 2025-09-15 | In Progress |
| 2 | Automate dependency scanning for Spring Framework versions | DevSecOps | 2025-10-01 | Planned |
| 3 | Enhance firewall logging to capture exploit payload details | Security Operations | 2025-09-01 | In Progress |
| 4 | Conduct postmortem tabletop on zero-day RCE exploits | Incident Response Lead | 2025-09-30 | Planned |
| 5 | Update IR playbook with Spring4Shell mitigation steps | SOC Manager | 2025-09-10 | Completed |

**8. Preventive Controls & Next Steps**

* **90-Day Roadmap:**
  1. Implement SBOM-driven patch alerts
  2. Expand WAF rule library to cover emerging exploits
  3. Deploy runtime application self-protection (RASP) on critical endpoints
* **Long-Term:**
  1. Integrate automated threat-pattern feeds into firewall and SIEM
  2. Biannual security training for developers and operators

**Conclusion:**  
Rapid cross-team collaboration and signature-based filtering contained this critical RCE exploit. Ongoing enhancements to patch automation, threat detection, and incident readiness will bolster resilience against future zero-day attacks.