Incident Postmortem: Spring4Shell Exploitation Attempt on Public Web Server

## **Summary**

SOC detected suspicious HTTP POST and GET requests targeting /index.jsp and /tomcatwar.jsp on 28th July at 1:05 AM IST (7:35 PM UST , 27th July). These requests matched known Spring4Shell patterns. The incident was categorized as High Severity due to the widespread and automated nature of the attack. The SOC team, Infrastructure Owner, and the Network Team were involved.

## **Impact**

No systems were compromised. However, this was a targeted distributed attack attempting to upload a .jsp web shell by manipulating internal Java class loader properties.The malicious traffic was blocked proactively. Normal services remained unaffected throughout the incident.

## **Detection**

Throughout the attack there were repeated access to /index.jsp and /tomcatwar.jsp. Spring4Shell exploits were detected. Malicious parameters were used.   
  
During firewall log analysis SOC alerts were triggered due to the suspicious activites.

## **Root Cause**

Spring4Shell vulnerability is the root cause through which attackers manipulate internal class loader to write malicious files.

## **Resolution**

New python based firewall was deployed. Malicious patterns were blocked using 403 Forbidden responses.   
  
An Attack was simulated that confirmed incident containment.

## **Action Items**

Following was done to resolve the incident:

Firewall server rule was implemented to detect and block Spring4Shell payloads.   
Communication within teams was completed.

Following needs to be done for the future:

Ensuring that updated version of Spring Boot is being used.  
WAF coverage needs to be applied throughout   
Regular simulations need to be conducted to prepare for any threat.  
Update incident response playbooks with Spring4Shell patterns.