**From:** Abdifatah Galgalo

**To:** Telstra Security Operations centre

**Subject:** Firewall Rule Creation & Ongoing Attack Analysis

Dear Telstra

After analyzing the firewall logs and reviewing the Spring Framework vulnerabilities, we identified multiple bypass actions or techniques. The incident occurred between 2022-03-20T03:16:34Z and 2022-03-20T03:21:00Z, allowing potentially malicious requests via the firewall. Below are the key findings. Also, the key infrastructure is currently under attack is NBN Connection. The found activity is critical of nature due to the below findings.

**Attack Details**

**Suspicious Request Paths:**

* **/tomcatwar.jsp** – Although this is a Tomcat-related path, the request data contains Java class loader references, a known **Spring4Shell exploit technique**.
* No direct requests to **/actuator**, **/env**, or **/gateway** were found, but the presence of **class.module.classLoader** suggests an attempt to exploit **Spring-based applications**.
* Header suffix=%>// c1=Runtime c2=<% DNT=1 Content-Type=application/x-www-form-urlencoded
  + This is likely an injection attack attempting to break out of a template or script context, introduce executable Java code, and possibly execute system commands via Runtime. If a web app processes this unsafely, it could lead to **Remote Code Execution (RCE)**.

* **suffix=%>//** → This might be trying to close an existing statement (%>) and then comment out the rest of the line using //.
* **c1=Runtime** → Could be referencing Java's Runtime class, which is often used to execute system commands.
* **c2=<%** → The <% is used in JSP to start a scriptlet, possibly attempting to inject code.
* **DNT=1** → A standard "Do Not Track" header, possibly included as noise or to evade detection.
* **Content-Type=application/x-www-form-urlencoded** → Indicates that this payload might be part of a form submission.

**Request Methods:**

* **All identified requests use HTTP POST**, which is commonly used for injecting **remote code execution (RCE) payloads**.

**Payload Analysis:**

* The clientRequestData contains **class.module.classLoader.resources.context**, a pattern seen in **Spring4Shell (CVE-2022-22965)** and similar RCE vulnerabilities.

**Type of Attack Identified: (Timestamp:** 2022-03-20T03:21:00Z )

☑ **Suspicious request data referencing Java class loaders**, commonly associated with **remote code execution (RCE) attacks**.

**Recommended Firewall Rules to Mitigate the Attack:**

☑ **Block all "bypass" actions**, ensuring that unauthorized traffic does not get through.

☑ **Monitor and restrict repeated attack attempts**, including requests that were initially blocked.

**Additional Research & Findings**

Further investigation into **Spring Framework security** confirms that this attack aligns with previously documented **Spring4Shell exploits**.

According to research from **Palo Alto Networks**, two major vulnerabilities were reported within the Spring Framework—an open-source framework used for building enterprise Java applications.

On **March 29, 2022**, the **Spring Cloud Expression Resource Access Vulnerability** (**CVE-2022-22963**) was patched in **Spring Cloud Function 3.1.7 and 3.2.3**.

For more details, refer to **Palo Alto's security analysis:**

🔗 [CVE-2022-22965 SpringShell Attack Analysis](https://unit42.paloaltonetworks.com/cve-2022-22965-springshell/)

Please let us know if you need further analysis or assistance in implementing the recommended firewall rules.

Best regards,

**Abdifatah Galgalo**

Telstra Security Operations Center