

Chasing Shell

A Log4Shell exploit on a DMZ server, contained in 12 minutes.

Incident Overview

Threat Type: Log4Shell RCE staging a cryptominer

Target: Internet-facing DMZ server

Time to Containment: 12 Minutes

The attacker sent a crafted request carrying a JNDI/LDAP lookup string to a vulnerable internet-facing Java application, forcing it to reach attacker-controlled infrastructure and run a Base64-encoded PowerShell download cradle. The script pulled a second-stage payload from a command-and-control (C2) server geolocated to Korea. CRITICALSTART®'s SOC decoded the multi-stage payload to expose the threat actor's full playbook.

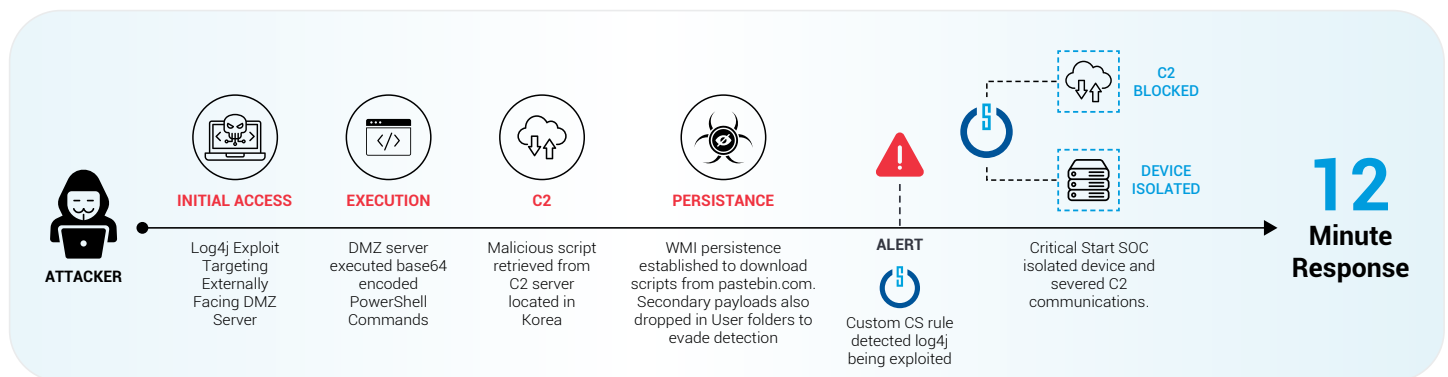
Detection & escalation

Detection: A custom Critical Start detection rule flagged the Log4j exploitation attempt in cloud SIEM telemetry that generic signatures would not have prioritized.

SOC Escalation: Escalated high priority. Analysts confirmed the JNDI exploit string, decoded the PowerShell cradle, and recognized an active code-execution chain rather than a routine scan — warranting immediate containment.

Depth of Investigation

Analysts identified Windows Management Instrumentation (WMI) event-subscription persistence firing every five minutes, scheduled-task fallback persistence, secondary scripts retrieved from a public paste site, and a cryptominer dropped under public user folders while disguised as a legitimate kernel process. This depth let the SOC map every persistence mechanism and indicator of compromise and deliver precise remediation guidance — including blocking all attacker infrastructure and rolling the host back to a known-good state.



Outcome

In 12 minutes, the Critical Start SOC isolated the affected server and cut off command-and-control communication, stopping the exploitation chain before the cryptominer could fully establish persistence or spread. The customer received clear next steps to patch the vulnerable application, block the attacker infrastructure, and restore the host with confidence.