Log4J Update
JAN 18 2022
Background

- Started in early 1996 as tracing API. After countless enhancements and several incarnations, the initial API has evolved to become log4j, a popular logging package for Java.

- The package is distributed under the Apache Software License and log4j has been ported to the C, C++, C#, Perl, Python, Ruby, and Eiffel languages.

- Log4j is used in everything from light switches to server farms to log activity ranging from HTTP headers to end user interaction with webforms.
Vulnerability History

- Security researcher Chen Zhaojun of Alibaba, first reported the vulnerability to the Apache Foundation on November 24.
- They discovered the attack December 9 on servers that host the game Minecraft.
- After further forensic analysis, they realized cybercriminals discovered the gap earlier, and have exploited it since at least December 1, 2021 making it a 0 day exploit.
- Attackers have been exploiting the vulnerability to compromise virtualization infrastructure, install and execute ransomware, steal system credentials, take broad control of compromised networks, and exfiltrate data.
Vulnerability Timeline

- Thursday, December 9: Apache Log4j zero-day exploit discovered. A fix for the issue was made available with the release of Log4j 2.15.0.
- Saturday, December 11: CISA director comments on “urgent challenge” recommending
  - Enumerate any external facing devices that have Log4j installed
  - Ensure security operations centers are actioning every single alert on the devices that fall into the category above
  - Install a web application firewall with rules that automatically update so that security operations centers (SOCs) can concentrate on fewer alerts
- Tuesday, December 14: Second Log4j vulnerability carrying denial-of-service threat detected, new patch 2.16.0 released
- Friday, December 17: Third Log4j vulnerability revealed, new fix made available
Friday December 10: InfoSec team begins talking to system owners informing them of the risk and asking to enumerate vulnerabilities.

Saturday December 11: CISA alert elevates Jeffco’s activities. About 35 staff members from all Jeffco IT teams, Communications, Student Data, and Ed Tech were brought together to begin Incident Response, MIM, and DR processes.

- Goals were to formalize enumeration efforts, define compensating controls, formalize vendor communication, create remediation/mitigation strategy.
- The team decided we did not have adequate compensating controls in place for external facing systems so we limited public access.

Sunday Dec 12 – Wednesday Dec 15: Goals were to safely bring core systems back online, continue to enumerate tertiary systems, continue to automate detection/containment systems.

Wednesday December 15: testing was complete on 2.15 patching while we discovered 2.16 was necessary so access to core systems continued to be limited.
Jeffco Timeline

- Friday December 17: 2.16 was applied to major systems and we felt comfortable with some cloud partner mitigation plans.

- **Other Facts**
  - Jeffco is managing through mitigations on 185 systems
  - Internal systems, cloud systems, and installed applications all offer unique challenges
  - We have found some vendors' mitigation plans are incorrect, so we are trusting but verifying
Campus Example

- Patching necessary to bring Campus back online:
  - Edge router
  - Campus application
  - Load balancer
  - Server operating systems
  - Backup software
  - VM software
  - Server management software
  - SAN management software
Questions?

- How did our strategy resonate with your community?
- Do you feel our actions were appropriate?
- Is there anything we could have done better?/What have your organizations done to manage through remediations?
- Has that trust continued as we worked through this weekend's outage?