Comprehensive Report on Addressing Emerging Vulnerabilities: Apache Log4j Zero-Day and Ransomware

Prepared by: Damian Mutisya, Information Security Analyst, Cyber & Information Security Team
Organization: Consultant AIG

---

1. Background Information

As an Information Security Analyst in the Cyber & Information Security Team, your role is critical in identifying, assessing, and mitigating vulnerabilities to safeguard AIG's infrastructure. Recently, the Cybersecurity & Infrastructure Security Agency (CISA) published advisories on two significant cybersecurity concerns:

1. **Apache Log4j Zero-Day Vulnerability:** This vulnerability affects one of the world's most widely used logging software components, posing a severe risk of remote code execution (RCE).
2. **Ransomware Professionalization:** This advisory highlights the increasing threat of ransomware, which is becoming more professionalized, sophisticated, and targeted, thereby posing a significant risk to large companies like AIG.

This report outlines the analysis of the recent Log4j vulnerability, identifies affected AIG infrastructure, and provides detailed recommendations for remediation.

---

2. Vulnerability Overview: Apache Log4j

Vulnerability: The Log4j vulnerability, also known as **CVE-2021-44228** and **CVE-2021-45046**, allows an unauthenticated attacker to perform remote code execution on affected versions of the software. This is a critical vulnerability due to its potential to enable attackers to gain control over compromised systems.

Affected Versions: Log4j versions from 2.0-beta9 to 2.15.0 are vulnerable.

Risk and Impact:

- **Severity:** Critical
- **Impact:** Remote Code Execution (RCE), which could lead to unauthorized access, data exfiltration, or execution of malicious actions on affected infrastructure.

Remediation Steps:

- **Identify Assets:** Determine all systems and infrastructure using the vulnerable versions of Log4j.
- **Update:** Upgrade to secure versions: Log4j 2.16.0 (Java 8) or 2.12.2 (Java 7).
- **Monitor:** Actively monitor for any signs of exploitation and respond promptly to any identified incidents.

---

3. Infrastructure Analysis and Affected Teams

Based on the provided infrastructure list, the Product Development Staging Environment managed by the Product Development Team is impacted by the Log4j vulnerability. The relevant details are as follows:

Product Name	Team Lead	Services Installed	Status
Product Development Staging Environment	John Doe (product@email.com)	Dovecot, pop3d, Apache httpd, **Log4j**, Dovecot imapd, MiniServ	**Affected**

---

4. Advisory Email to Affected Team

From: AIG Cyber & Information Security Team
To: Product Development Team (product@email.com)
Subject: Security Advisory Concerning Product Development Staging Environment | Log4j

---

Body:

Hello John Doe,

The AIG Cyber & Information Security Team would like to inform you that a critical vulnerability has been discovered in the Apache Log4j software that may affect the Product Development Staging Environment infrastructure.

Vulnerability Overview
Log4j is a widely used open-source logging framework. The identified vulnerability (CVE-2021-44228 and CVE-2021-45046) allows an attacker to execute remote code on affected systems, potentially compromising the environment.

Affected Products
Log4j versions 2.0-beta9 through 2.15.0

Risk & Impact

- **Severity:** Critical
- **Impact:** Remote code execution (RCE) could lead to unauthorized access, data exfiltration, or execution of malicious actions on the Product Development Staging Environment.

Remediation

- **Identify Assets:** Review and identify all systems utilizing vulnerable versions of Log4j.
- **Update:** Upgrade to secure versions: Log4j 2.16.0 (Java 8) or 2.12.2 (Java 7).
- **Monitor:** Be vigilant for any exploitation attempts and notify the security team immediately if any are detected.

After remediating this vulnerability, please confirm the actions taken by replying to this email. For any questions or further assistance, feel free to reach out to us.

Kind regards,

AIG Cyber & Information Security Team

---

5. Conclusion and Recommendations

The Apache Log4j vulnerability represents a significant risk due to its widespread use and the critical nature of the vulnerability. Immediate action is required to mitigate this risk across affected AIG infrastructure. In addition to addressing this specific vulnerability, it is recommended to:

- **Enhance Monitoring:** Implement enhanced monitoring across all environments to detect signs of exploitation early.
- **Update Incident Response Plans:** Ensure that incident response plans are updated to reflect the increased risks posed by zero-day vulnerabilities and ransomware threats.
- **Conduct Regular Security Audits:** Regularly audit and patch systems to minimize exposure to similar vulnerabilities in the future.

Report 2: Bypassing Ransomware by Bruteforcing the Decryption Key

---

Prepared by: Damian Mutisya, Information Security Analyst, Cyber & Information Security Team
Organization: Consultant AIG

Overview

Following a successful exploitation of the Log4j vulnerability, an attacker attempted to deploy ransomware on the Product Development Staging Environment. Although the ransomware failed to fully encrypt the server, it did manage to encrypt a single ZIP file. This report details the approach taken to bruteforce the decryption key without paying the ransom.

Objective

To recover access to the encrypted ZIP file (`enc.zip`) without paying the ransom by developing a Python script to bruteforce the decryption key using a subset of common passwords from the Rockyou wordlist.

Technical Approach

1. **Open the Encrypted ZIP File:** The script reads the encrypted ZIP file in read mode.
2. **Read the Password Wordlist:** A subset of common passwords from the Rockyou wordlist is used for bruteforce attempts.
3. **Bruteforce Attempts:** The script iterates over each password, attempting to extract the ZIP file using the current password.
4. **Logging:** Outputs the password attempts in real-time and indicates if the correct password is found.

Python Code Implementation

```
from zipfile import ZipFile
import zipfile

def attempt_extract(zf_handle, password):
    """
    Attempt to extract the contents of the zip file using the provided password.

    :param zf_handle: The zip file handle
    :param password: The password to test
    :return: True if extraction is successful, False otherwise
    """
    try:
        # Encode password to bytes as required by extractall
        zf_handle.extractall(pwd=password.encode('latin-1'))
        # If extraction is successful, return True
        return True
    except (zipfile.BadZipFile, RuntimeError):
        # Return False if extraction fails due to incorrect password or other
issues
        return False

def main():
    """
    Main function to handle the bruteforce attack on the encrypted zip file.
    """
    print("[+] Beginning bruteforce ")
    # Open the encrypted zip file in read mode
    with ZipFile('enc.zip', 'r') as zf:
        # Open the password list file
        with open('rockyou.txt', 'rb') as f:
            # Iterate over each password in the list
            for line in f:
                # Decode the password using latin-1 to handle common encoding
issues
                password = line.strip().decode('latin-1')
                # Print the current password attempt
                print(f"[*] Trying password: {password}")
                # Try to extract the zip file using the current password
                if attempt_extract(zf, password):
                    # Print the found password and exit on success
                    print(f"[+] Password found: {password}")
                    return
    # If no password is found in the list, print the failure message
```

```
    print("[+] Password not found in list")

if __name__ == "__main__":
    main()
```

Execution and Results

Execution Instructions:

1. Ensure that `enc.zip` and `rockyou.txt` are in the same directory as the script.
2. Run the script using Python 3:

   bash

```
    python3 bruteforce_ransomware.py
```

Results:

- The script attempts each password from the list to unlock the encrypted ZIP file.
- If the correct password is found, it is printed to the console.
- If no password is found in the list, a message indicates the failure.

Conclusion and Recommendations

The bruteforce method effectively recovered access to the encrypted file without paying the ransom. This incident highlights the importance of robust incident response strategies and the need for continuous monitoring of vulnerabilities.

Recommendations:

- Strengthen incident response protocols.
- Implement comprehensive monitoring for signs of exploitation.
- Regularly back up critical data.
- Enforce strong password policies and encourage the use of multi-factor authentication.