Task Overview

**What you'll learn**

* How to address a vulnerability that may affect Product Development Staging Environment infrastructure

**What you'll do**

* Review some recent publications from the Cybersecurity & Infrastructure Security Agency (CISA)
* Research the reported vulnerability
* Draft an email to affected teams to alert them of the vulnerability, and explain how to remediate

Here is the background information for your task

You are an **Information Security Analyst** in the Cyber & Information Security Team.

A common task and responsibility of information security analysts is to stay on top of emerging vulnerabilities to make sure that the company can remediate them before an attacker can exploit them.

In this task, you will be asked to review some recent publications from the Cybersecurity & Infrastructure Security Agency (CISA). The Cybersecurity & Infrastructure Security Agency (CISA) is an Agency that has the goal of reducing the nation’s exposure to [cyber](https://www.theforage.com/virtual-experience/2ZFnEGEDKTQMtEv9C/aig/cybersecurity-ku1i/www.google.com) security threats and risks.

After reviewing the publications, you will then need to draft an email to inform the relevant infrastructure owner at AIG of the seriousness of the vulnerability that has been reported.

**Task 1: Responding to a zero-day vulnerability**

Here are the instructions for your task

The CISA has recently published the following two advisories:

1. The [first advisory (Log4j)](https://www.cisa.gov/uscert/ncas/alerts/aa21-356a), outlines a serious vulnerability in one of the world’s most popular logging software.
2. The [second advisory](https://www.cisa.gov/news/2022/02/09/cisa-fbi-nsa-and-international-partners-issue-advisory-ransomware-trends-2021) explores how ransomware has been increasing and is becoming professionalized - a concern for a large company like AIG.

Your task is to respond to the Apache Log4j zero-day vulnerability that was released to the public by advising affected teams of the vulnerability.

**First,**conduct your research on the vulnerability using the “CISA Advisory" resources provided above as a starting point.

**Next,**analyze the “Infrastructure List” below to find out which infrastructure may be affected by the vulnerability, and which team has ownership.

|  |  |  |  |
| --- | --- | --- | --- |
| **Product Team** | **Product Name** | **Team Lead** | **Services Installed** |
| IT | Workstation Management System | Jane Doe (tech@email.com) | OpenSSH dnsmasq lighttpd |
| Product Development | Product Development Staging Environment | John Doe (product@email.com) | Dovecot pop3d Apache httpd Log4j Dovecot imapd MiniServ |
| Marketing | Marketing Analytics Server | Joe Schmoe (marketing@email.com) | Microsoft ftpd Indy httpd Microsoft Windows RPC Microsoft Windows netbios-ssn Microsoft Windows Server 2008 R2 - 2012 microsoft ds |
| HR | Human Resource Information System | Joe Bloggs (hr@email.com) | OpenSSH Apache httpd rpcbind2-4 |

Draft your advisory email below

To finish this task, draft an advisory email to alert the infrastructure owner of the seriousness of this vulnerability.

For inspiration, you can use the email template provided below from our last cyber threat advisory.

**From:**AIG Cyber & Information Security Team  
**To:**<affected team>  
**Subject:**Security Advisory concerning <affected product> <affected software>  
—  
**Body:**  
Hello <affected team owner>,

AIG Cyber & Information Security Team would like to inform you that a recent <affected software> vulnerability has been discovered in the security community that may affect <affected product>.

<vulnerability description>

<vulnerability risk/impact>

<vulnerability remediation>

<any assurances to ensure advisory was actioned>

For any questions or issues, don’t hesitate to reach out to us.

Kind regards,  
AIG Cyber & Information Security Team

Tips for your email:

* Make it direct and straight to the point.
* You can assume the infrastructure owner is technical.
* Explain the risk/impact, method of exploitation, and remediation steps.

**From:** AIG Cyber & Information Security Team  
**To:** John Doe (product@email.com)  
**Subject:** Security Advisory concerning Product Development Staging Environment - Log4j  
—  
**Body:**  
Hello John,

The AIG Cyber & Information Security Team would like to inform you of a critical vulnerability recently discovered in the Apache Log4j logging framework that affects the Product Development Staging Environment.

**vulnerability description**

The Log4j vulnerability (CVE-2021-44228) allows unauthenticated remote code execution (RCE) by sending a specially crafted request to a system using Log4j. This exploit leverages the framework's logging feature to inject malicious code, potentially compromising the affected system.

**risk/impact**

This vulnerability is classified as critical (CVSS score: 10.0) and poses severe risks, including:

* Full system compromise.
* Unauthorized access to sensitive data.
* The possibility of the staging environment being leveraged as a pivot point for further attacks within the infrastructure.

**vulnerability remediation**

Immediate action is required to mitigate this vulnerability:

1. **Update Log4j**: Upgrade to version 2.15.0 or later, where the vulnerability has been patched.
2. **Temporary Mitigation (if updating is not immediately possible)**:
   * Set the log4j2.formatMsgNoLookups system property to true.
   * Remove the JndiLookup class from the classpath by running:

zip -q -d log4j-core-\*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

1. Review and monitor logs for any signs of exploitation.
2. Conduct a thorough security assessment of the affected environment post-remediation.

**Action Required**

Please confirm the completion of the above steps or share any challenges you encounter in addressing this issue.

For any questions or further assistance, don’t hesitate to reach out to us.

Kind regards,  
AIG Cyber & Information Security Team

Example advisory email

Great work!

There are many ways you could have attempted this task, as advisory emails come in all shapes and sizes. Below you'll find one example of an advisory email alerting the infrastructure owner of the seriousness of this vulnerability.

**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,  
  
AIG Cyber & Information Security Team would like to inform you that a recent Log4j vulnerability has been discovered in the security community that may affect the Product Development Staging Environment infrastructure.  
  
Vulnerability Overview  
Log4j is a common open-source tool used for application logging and monitoring across the web. Recently, a vulnerability has been identified in versions Log4j2 2.0-beta9 through 2.15.0 that would allow an unauthenticated attacker to perform remote code execution on affected infrastructure, making this a critical vulnerability. You can learn more in the NIST disclosures: [NVD - CVE-2021-44228](https://nvd.nist.gov/vuln/detail/CVE-2021-44228) and [NVD - CVE-2021-45046](https://nvd.nist.gov/vuln/detail/CVE-2021-45046).  
  
Affected products  
Log4j2 2.0-beta9 through 2.15.0  
  
Risk & Impact  
Critical - remote code execution (RCE). An attacker will be able to remotely access the Product Development Staging Environment infrastructure to exfiltrate data or execute malicious actions.  
  
Remediation  
● Identify any assets or infrastructure running the affected Log4j version  
● Update to the following versions: Log4j 2.16.0 (Java 8) and 2.12.2 (Java 7)  
● Be on the lookout for any signs of exploitation  
  
If you identified any signs of exploitation, please immediately reach out. After you have remediated this vulnerability, please confirm with the security team by replying to this email.  
  
For any questions or issues, don’t hesitate to reach out to us.  
  
Kind regards,  
AIG Cyber & Information Security Team

**Task 2: (Technical) Bypassing ransomware**

Task Overview

**What you'll learn**

* What ‘bruteforcing’ involves
* How to respond to a ransomware virus using Python

**What you'll do**

* Write a Python script to bruteforce the decryption key of the encrypted file, to avoid paying a ransom

Setting the scene for your next task

Your advisory email in the last task was great. It provided context to the affected teams on what the vulnerability was, and how to remediate it.

Unfortunately, an attacker was able to exploit the vulnerability on the affected server and began installing a ransomware virus. Luckily, the Incident Detection & Response team was able to prevent the ransomware virus from completely installing, so it only managed to encrypt one zip file.

Internally, the Chief Information Security Officer does not want to pay the ransom, because there isn’t any guarantee that the decryption key will be provided or that the attackers won’t strike again in the future.

Instead, we would like you to bruteforce the decryption key. Based on the attacker’s sloppiness, we don’t expect this to be a complicated encryption key, because they used copy-pasted payloads and immediately tried to use ransomware instead of moving around laterally on the network.

**Here is the background information for your task**

In this task, you will write a Python script to bruteforce the decryption key of the encrypted file.

Bruteforcing is the act of repeatedly trying different combinations to break the password encryption (based on either randomly generated passwords, or from a list of passwords to try). In the resource below, we've provided a small subset of passwords from Rockyou - a widely know password wordlist that contains thousands of common passwords in one wordlist.

Ransomware will often encrypt all files on a device, and sometimes give the decryption key after the ransom has been paid (but this is not always the case!). In this task, we would like you to break the encryption without paying the ransom.

A foundational Python 3+ template has also been provided for you in the resource below. One potential implementation is described in the code comments.

The python code for brute force attack is written below:

'''

Forage AIG Cybersecurity Program

Bruteforce starter template

'''

from zipfile import ZipFile

def attempt\_extract(zf\_handle, password):

try:

zf\_handle.extractall(pwd=password.encode('utf-8'))

return True

except Exception as e:

return False

def main():

print("[+] Beginning brute force attack")

with ZipFile('enc.zip') as zf:

with open('rockyou.txt', 'r', encoding='utf-8', errors='ignore') as f:

for line in f:

password = line.strip() # Remove any leading/trailing whitespace

print(f"[-] Trying password: {password}")

if attempt\_extract(zf, password):

print(f"[+] Success! Password found: {password}")

return

print("[-] Password not found in list")

if \_\_name\_\_ == "\_\_main\_\_":

main()