**Khai thác lỗ hổng Jackson Deserialize Vulnerabilities - CVE-2017-7525**

*Presented by Le Van Luong*

Bài này được thực hiện trên môi trường java 8, thư viện jackson 2.6.1

## 1, Khái quát về thư viện jackson

Là thư viện java phổ biến xử lý json, dùng để chuyển đổi qua lại (serialize/deserialize) dữ liệu giữa Java objects và JSON

## 2, Lỗ hổng deserialize trong jackson

Tận dụng cơ chế Polymorphic Type Handling (xử lý kiểu đa hình)

Giải thích về cách hoạt động của PTH

Giả sử ta có chương trình sau:  
A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

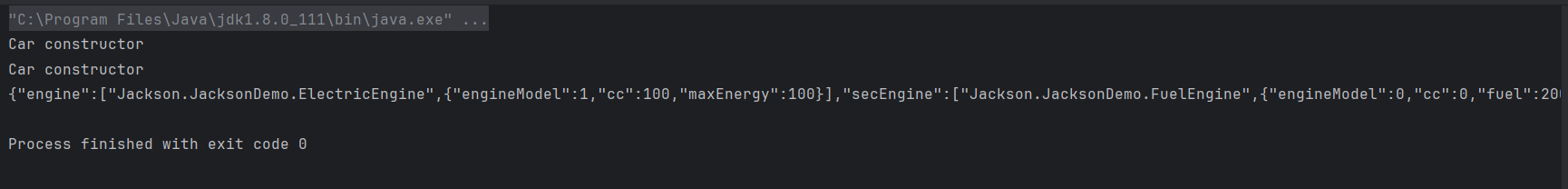
AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Kết quả: khi serialize thì bình thường nhưng khi thử deserialize, ta nhận được:  
A computer screen shot of text

AI-generated content may be incorrect.

Jackson không thể khởi tạo 1 đối tượng của lớp trừu tượng Engine khi deserialize JSON thành đối tượng Car . Nguyên nhân do jackson không biết chính xác cần khởi tạo lớp con nào của thuộc tính secEngine trong Car. Để giải quyết vấn đề này **CÓ THỂ** sử dụng phương thức objectMapper.enableDefaultTyping(), thử chạy lại:  


Đây là 1 trong các cách xử lý kiểu đa hình của Jackson. Tuy nhiên đối với việc sử dụng phương thức enableDafaultTyping() này tuy đơn giản và thuận tiện nhưng nguy hiểm đối với các đối tượng Java có các kiểu như java.lang.Object, java.util.Serializable, java.util.Comparable là nguy hiểm vì nhiều đối tượng Java là các kiểu con của các kiểu này.

Thử thay đổi nội dung file json thành như sau:  
A screenshot of a computer program

AI-generated content may be incorrect.

A close-up of a web page

AI-generated content may be incorrect.

Vì ta khai báo thuộc tính engine là Object (là lớp cha của tất cả các lớp trong java) nên jackson có thể deserialize bất cứ kiểu dữ liệu nào vào, bao gồm FileSystemXMLApplicationContext.

A screen shot of a computer program

AI-generated content may be incorrect.

File spel.xml: File XML này chứa định nghĩa các bean Spring, bao gồm:

Một bean "exploit" thuộc lớp java.lang.ProcessBuilder, được cấu hình để chạy lệnh "calc.exe".

Một bean "trigger" thuộc lớp org.springframework.beans.factory.config.MethodInvokingFactoryBean, được cấu hình để gọi phương thức "start" trên bean "exploit", kích hoạt quá trình chạy lệnh.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Mấu chốt là gọi được setter hoặc constructor của biến engine (kiểu object). Có thể khai báo thuộc tính engine là private hoăc public đều được. nếu public thì khi gọi constructor của car thì nó tự động gọi được luôn, còn private thì phải viết hàm set cho nó để jackson còn gán thông qua đó. Chạm được đến lớp Object thì FileSystemXmlApplicationContext mới được khởi tạo => gadget chain:

Jackson ObjectMapper

-> Car.engine (Object)

-> FileSystemXmlApplicationContext (constructor với configLocation)

-> XML Bean Definition (spel.xml)

-> ProcessBuilder (start())

-> Command Execution

org.springframework.context.support.FileSystemXmlApplicationContext: Đây là một lớp tiện ích của Spring để tạo context từ file XML. Khi được khởi tạo với URL, nó tự động tải và phân tích file XML, tạo các bean được định nghĩa. Trong trường hợp này, nó tải spel.xml, dẫn đến tạo và kích hoạt các bean nguy hiểm.

java.lang.ProcessBuilder: Lớp này cho phép tạo một tiến trình mới bằng cách chỉ định lệnh và các tham số. Trong trường hợp này, nó được cấu hình để chạy "calc.exe".

org.springframework.beans.factory.config.MethodInvokingFactoryBean: Lớp này thuộc Spring Framework, được sử dụng để gọi phương thức trên một bean khác. Trong spel.xml, nó được cấu hình để gọi phương thức "start" trên bean ProcessBuilder, đảm bảo lệnh được thực thi khi context được làm mới.

Debug:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

vanillaDeserialize trong BeanDeserializer của Jackson xử lý việc giải mã JSON cơ bản, đặc biệt khi không có tính năng đặc biệt nào được kích hoạt, và có thể dẫn đến tạo đối tượng tùy ý khi default typing được bật

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Các điều kiện để gây ra lỗ hổng jackson rce:

* Thư viện jackson <= 2.6.1
* Sử dụng ObjectMapper.enableDefaultTyping
* Có thuộc tính có kiểu Object được khởi tạo (có phương thức set / constructor và được gọi đến)
* Thiếu sót trong việc xử lý đầu vào tệp json

Biện pháp giảm thiểu:

1. Tắt tính năng Polymorphic Deserialization nếu không cần thiết hoặc cấu hình ObjectMapper để giới hạn các lớp được phép deserialize
2. Sử dụng Whitelist cho các lớp được deserialize
3. Tránh deserialize các thuộc tính không khớp với đối tượng đích
4. Tránh sử dụng Object hoặc các kiểu Generic quá rộng
5. Kiểm tra và lọc dữ liệu đầu vào
6. Cập nhật phiên bản jackson mới nhất
7. Có thể giám sát và ghi log để phát hiện các hành vi deserialize bất thường