羊城杯2021-网络安全大赛部分writeup

原创

KogRow 于 2021-09-11 22:09:18 发布 1631 收藏 2

分类专栏： CTF apk逆向 杂项 文章标签： 网络安全 CTF

本文链接：https://blog.csdn.net/shuaicenglou3032/article/details/120233576

1.签到

```php
<?php
echo("SangFor{".md5("28-08-30-07-04-20-02-17-23-01-12-19")."}");
?>
```

图1是二八定律（28），图2是八卦（8），图3是三十而立（30），图4是北斗七星（07），图5是江南四大才子（04），图6是歼20（20），图7是两个黄鹂（02），图8是一起来看流星雨（一起谐音17），图9是乔丹的球服（23），图10是一马当先（01），图11是十二星座（12），图12是19点播放的新闻联播（19）

2.Ez_android

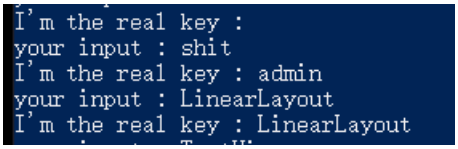
反编译得到核心代码：

```
package top.zjax.login;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
```

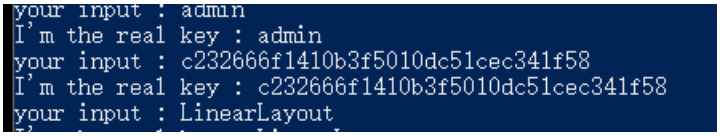
```
import java.io.OutputStreamWriter;
import java.math.BigInteger;
import java.net.Socket;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MainActivity extends Activity implements View.OnClickListener {
    String key = "";
    Button loginBtn = null;
    EditText passEt = null;
    TextView promptText = null;
    EditText useridEt = null;

    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(C0818R.layout.activity_main);
        Button button = (Button) findViewById(C0818R.C0821id.loginBtn);
        this.loginBtn = button;
        button.setOnClickListener(this);
        this.useridEt = (EditText) findViewById(C0818R.C0821id.userId);
        this.passEt = (EditText) findViewById(C0818R.C0821id.pass);
        this.promptText = (TextView) findViewById(C0818R.C0821id.promptText);
    }

    public void onClick(View view) {
        String trim = this.useridEt.getText().toString().trim();
        String trim2 = this.passEt.getText().toString().trim();
        if (trim.equals("")) {
            this.promptText.setText(C0818R.string.userIdError);
        } else if (trim2.equals("")) {
            this.promptText.setText(C0818R.string.passError);
        } else if (!checkUsername(trim) || !checkPasswd(trim2)) {
            Toast.makeText(this, (int) C0818R.string.loginError, 1).show();
        } else {
            Toast.makeText(this, (int) C0818R.string.loginSuccess, 1).show();
            getKeyAndRedirect(trim2);
        }
    }

    private boolean checkUsername(String str) {
        return str.equals(getString(C0818R.string.username));
    }

    private boolean checkPasswd(String str) {
        return getEncodeStr(str).equals(getString(C0818R.string.passwd));
    }

    private String getEncodeStr(String str) {
        byte[] bArr = null;
        try {
            bArr = MessageDigest.getInstance(getString(C0818R.string.encode)).digest(str.getBytes(StandardCharse
ts.UTF_8));
            for (int i = 0; i < bArr.length; i++) {
                bArr[i] = (byte) (bArr[i] - 1);
            }
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
```

```
        return new BigInteger(1, bArr).toString(16);
    }

    private void getKeyAndRedirect(String str) {
        new Thread(new Runnable(str) {
            /* class top.zjax.login.lambda */
            public final /* synthetic */ String f$1;

            {
                this.f$1 = r2;
            }

            public final void run() {
                MainActivity.this.lambda$getKeyAndRedirect$0$MainActivity(this.f$1);
            }
        }).start();
        while (this.key.length() == 0) {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        Intent intent = new Intent(this, CheckFlagActivity.class);
        intent.putExtra("key", this.key);
        startActivity(intent);
        finish();
    }

    public /* synthetic */ void lambda$getKeyAndRedirect$0$MainActivity(String str) {
        try {
            Socket socket = new Socket("139.224.191.201", 20080);
            BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()))
;
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            bufferedWriter.write(str);
            bufferedWriter.newLine();
            bufferedWriter.flush();
            setKey(bufferedReader.readLine().split(":")[1].trim());
            bufferedReader.close();
            bufferedWriter.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void setKey(String str) {
        this.key = str;
    }
}
```

使用frida去hook题目程序的equals方法得到用户名是admin：



同样的方法，hook到密码的比较，得到一串字符串 “c232666f1410b3f5010dc51cec341f58”：



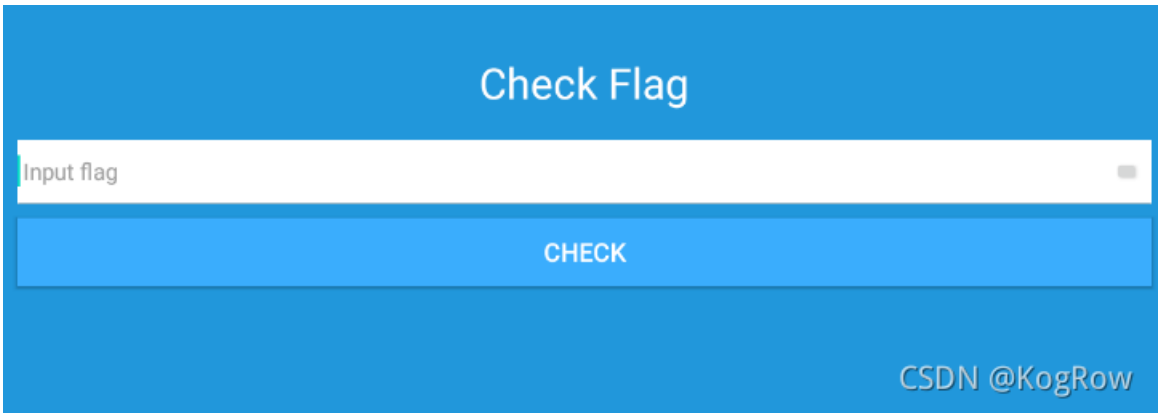
然后密码可以看到校验的方法是这样的：

```
private boolean checkPasswd(String str) {
        return getEncodeStr(str).equals(getString(C0818R.string.passwd));
    }
    private String getEncodeStr(String str) {
        byte[] bArr = null;
        try {
            bArr = MessageDigest.getInstance(getString(C0818R.string.encode)).digest(str.getBytes(StandardCharse
ts.UTF_8));
            for (int i = 0; i < bArr.length; i++) {
                bArr[i] = (byte) (bArr[i] - 1);
            }
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return new BigInteger(1, bArr).toString(16);
    }
```

这段代码是把传进来的密码进行md5计算摘要，得到的摘要是byte数组的形式，然后每个数组元素值-1，再转成16进制字符串，根据代码逻辑写逆推代码如下：

```java
package Main;
import java.io.UnsupportedEncodingException;
import java.math.BigInteger;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class Main {
 public static void main(String[] args) throws UnsupportedEncodingException {
  System.out.println(getEncodeStr("fuck"));
  System.out.println(getDecodeStr("c232666f1410b3f5010dc51cec341f58"));
 }
 public static String getDecodeStr(String str) {
  BigInteger b = new BigInteger(str, 16);
  byte[] bArr = b.toByteArray();
  for (int i = 1; i < bArr.length; i++) {
   System.out.println(bArr[i]);
            bArr[i] = (byte) (bArr[i] + 1);
        }
  return new BigInteger(1, bArr).toString(16);
 }
 public static String getEncodeStr(String str) throws UnsupportedEncodingException {
        byte[] bArr = null;
        try {
            bArr = MessageDigest.getInstance("MD5").digest(str.getBytes(StandardCharsets.UTF_8));
            System.out.println(new BigInteger(1, bArr).toString(16));
//          System.out.println(new BigInteger(1, bArr).toString(16));
            for (int i = 0; i < bArr.length; i++) {
             System.out.println(bArr[i]);
                bArr[i] = (byte) (bArr[i] - 1);
            }
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
//        System.out.println(new String(bArr));
        return new BigInteger(1, bArr).toString(16);
    }

}
```

执行代码得到密码的md5值：c33367701511b4f6020ec61ded352059，一查得到明文为654321。
使用admin/654321登录，进入佛莱格校验流程：



查看佛莱格校验的代码：

```java
package top.zjax.login;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import java.nio.charset.StandardCharsets;
public class CheckFlagActivity extends Activity implements View.OnClickListener {
    Button flagBtn = null;
    EditText flagTx = null;
    String key;
    TextView promptText = null;

    /* access modifiers changed from: protected */
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(C0818R.layout.activity_checkflag);
        Button button = (Button) findViewById(C0818R.C0821id.flagBtn);
        this.flagBtn = button;
        button.setOnClickListener(this);
        this.flagTx = (EditText) findViewById(C0818R.C0821id.flagTx);
        this.promptText = (TextView) findViewById(C0818R.C0821id.promptText);
        this.key = getIntent().getStringExtra("key");
    }

    public void onClick(View view) {
        String trim = this.flagTx.getText().toString().trim();
        if (trim.equals("")) {
            this.promptText.setText(C0818R.string.flagError);
        } else if (checkFlag(trim)) {
            Toast.makeText(this, (int) C0818R.string.flagSuccess, 1).show();
        } else {
            Toast.makeText(this, (int) C0818R.string.flagError, 1).show();
        }
    }

    private boolean checkFlag(String str) {
        return new String(EncodeUtils.encode(str.getBytes(StandardCharsets.UTF_8), false, this.key.getBytes(Stan
dardCharsets.UTF_8))).equals(getString(C0818R.string.encodeFlag));
    }
}
```

得知首先从一个IP地址那里取得一个key，然后根据这个key和输入的flag生成一个新的字符串，把这个字符串与encode_flag进行比较。

继续hook得到encode_flag=3lkHi9iZNK87qw0p6U391t92qlC5rwn5iFqyMFDl1t92qUnL6FQjqln76l-P：

```
I'm the real key :
your input : iK-gR2==
I'm the real key : 3lkHi9iZNK87qw0p6U391t92qlC5rwn5iFqyMFDl1t92qUnL6FQjqln76l-P
your input : LinearLayout
```

然后读取key的代码如下：

```
public static void main(String[] args) {
 try {
            Socket socket = new Socket("139.224.191.201", 20080);
            BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()))
;
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            bufferedWriter.write("654321");
            bufferedWriter.newLine();
            bufferedWriter.flush();
            String keyString = bufferedReader.readLine().split(":")[1].trim(); //TGtUnkaJD0frq61uCQYw3-FxMiRvNOB
/EWjgVcpKSzbs8yHZ257X9LldIeh4APom
            bufferedReader.close();
            bufferedWriter.close();
            socket.close();
            String encodeFlag = "3lkHi9iZNK87qw0p6U391t92qlC5rwn5iFqyMFDl1t92qUnL6FQjqln76l-P";
        } catch (IOException e) {
            e.printStackTrace();
        }
}
```

运行得到key:TGtUnkaJD0frq61uCQYw3-FxMiRvNOB/EWjgVcpKSzbs8yHZ257X9LldIeh4APom
然后我们看看这个key和输入的flag生成一个新的字符串的方法：

```
public static byte[] encode(byte[] bArr, boolean z, byte[] bArr2) {
        if (bArr == null) {
            return null;
        }
        int length = bArr.length;
        int i = 0;
        if (length == 0) {
            return new byte[0];
        }
        int i2 = (length / 3) * 3;
        int i3 = length - 1;
        int i4 = ((i3 / 3) + 1) << 2;
        int i5 = i4 + (z ? ((i4 - 1) / 76) << 1 : 0);
        byte[] bArr3 = new byte[i5];
        int i6 = 0;
        int i7 = 0;
        int i8 = 0;
        while (i6 < i2) {
            int i9 = i6 + 1;
            int i10 = i9 + 1;
            int i11 = ((bArr[i6] & 255) << 16) | ((bArr[i9] & 255) << 8);
            int i12 = i10 + 1;
            int i13 = i11 | (bArr[i10] & 255);
            int i14 = i7 + 1;
            bArr3[i7] = bArr2[(i13 >>> 18) & 63];
            int i15 = i14 + 1;
            bArr3[i14] = bArr2[(i13 >>> 12) & 63];
            int i16 = i15 + 1;
            bArr3[i15] = bArr2[(i13 >>> 6) & 63];
            i7 = i16 + 1;
            bArr3[i16] = bArr2[i13 & 63];
            if (z && (i8 = i8 + 1) == 19 && i7 < i5 - 2) {
                int i17 = i7 + 1;
                bArr3[i7] = 13;
                i7 = i17 + 1;
                bArr3[i17] = 10;
                i8 = 0;
            }
            i6 = i12;
        }
        int i18 = length - i2;
        if (i18 > 0) {
            int i19 = (bArr[i2] & 255) << 10;
            if (i18 == 2) {
                i = (bArr[i3] & 255) << 2;
            }
            int i20 = i19 | i;
            bArr3[i5 - 4] = bArr2[i20 >> 12];
            bArr3[i5 - 3] = bArr2[(i20 >>> 6) & 63];
            bArr3[i5 - 2] = i18 == 2 ? bArr2[i20 & 63] : 61;
            bArr3[i5 - 1] = 61;
        }
        return bArr3;
    }
```

这段代码其实就是base64，只不过它可以根据传入的码表生成base64编码的字符串，而这里传入的码表就是key.
所以根据上述信息，找一段自定义码表的代码来逆推：

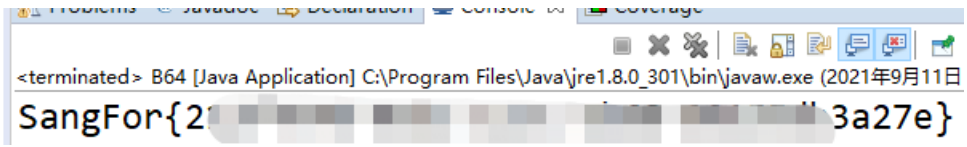
```
package Main;
```

```
public class B64 {



//     private static final char S_BASE64CHAR[] = {
//         'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
//         'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
//         'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd',
//         'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
//         'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x',
//         'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7',
//         '8', '9', '+', '/'
//     };
  //TGtUnkaJD0frq61uCQYw3-FxMiRvNOB/EWjgVcpKSzbs8yHZ257X9LldIeh4APom
   private static final char S_BASE64CHAR[] = "TGtUnkaJD0frq61uCQYw3-FxMiRvNOB/EWjgVcpKSzbs8yHZ257X9LldIeh4APom"
.toCharArray();




    private static final char S_BASE64PAD = 61;

    private static final byte S_DECODETABLE[];

    static {
        S_DECODETABLE = new byte[128];
        for(int i = 0; i < S_DECODETABLE.length; i++)
            S_DECODETABLE[i] = 127;

        for(int j = 0; j < S_BASE64CHAR.length; j++)
            S_DECODETABLE[S_BASE64CHAR[j]] = (byte)j;
    }


    private B64() { }


    public static byte[] decode(String s) {
        char ac[] = new char[4];
        int i = 0;
        byte abyte0[] = new byte[(s.length() / 4) * 3 + 3];
        int j = 0;
        for(int k = 0; k < s.length(); k++) {
            char c = s.charAt(k);
            if(c == '=' ||
               c < S_DECODETABLE.length &&
               S_DECODETABLE[c] != 127)
            {
                ac[i++] = c;
                if(i == ac.length) {
                    i = 0;
                    j += decode0(ac, abyte0, j);
                }
            }
        }

        if(j == abyte0.length) {
            return abyte0;
        } else {
```

```
            byte abyte1[] = new byte[j];
            System.arraycopy(abyte0, 0, abyte1, 0, j);
            return abyte1;
        }
    }


    public static String encode(byte abyte0[]) {
        return encode(abyte0, 0, abyte0.length);
    }


    public static String encode(byte abyte0[], int i, int j) {
        if(j <= 0)
            return "";

        char ac[] = new char[(j / 3) * 4 + 4];
        int k = i;
        int l = 0;
        int i1;
        for(i1 = j - i; i1 >= 3; i1 -= 3) {
            int j1 = ((abyte0[k] & 0xff) << 16) + ((abyte0[k + 1] & 0xff) << 8) + (abyte0[k + 2] & 0xff);
            ac[l++] = S_BASE64CHAR[j1 >> 18];
            ac[l++] = S_BASE64CHAR[j1 >> 12 & 0x3f];
            ac[l++] = S_BASE64CHAR[j1 >> 6 & 0x3f];
            ac[l++] = S_BASE64CHAR[j1 & 0x3f];
            k += 3;
        }

        if(i1 == 1) {
            int k1 = abyte0[k] & 0xff;
            ac[l++] = S_BASE64CHAR[k1 >> 2];
            ac[l++] = S_BASE64CHAR[k1 << 4 & 0x3f];
            ac[l++] = '=';
            ac[l++] = '=';
        } else if(i1 == 2) {
            int l1 = ((abyte0[k] & 0xff) << 8) + (abyte0[k + 1] & 0xff);
            ac[l++] = S_BASE64CHAR[l1 >> 10];
            ac[l++] = S_BASE64CHAR[l1 >> 4 & 0x3f];
            ac[l++] = S_BASE64CHAR[l1 << 2 & 0x3f];
            ac[l++] = '=';
        }
        return new String(ac, 0, l);
    }


    private static int decode0(char ac[], byte abyte0[], int i) {
        byte byte0 = 3;
        if(ac[3] == '=')
            byte0 = 2;
        if(ac[2] == '=')
            byte0 = 1;
        byte byte1 = S_DECODETABLE[ac[0]];
        byte byte2 = S_DECODETABLE[ac[1]];
        byte byte3 = S_DECODETABLE[ac[2]];
        byte byte4 = S_DECODETABLE[ac[3]];
        switch(byte0) {
        case 1: // '\001'
            abyte0[i] = (byte)(byte1 << 2 & 0xfc | byte2 >> 4 & 3);
```

```
            return 1;

        case 2: // '\002'
            abyte0[i++] = (byte)(byte1 << 2 & 0xfc | byte2 >> 4 & 3);
            abyte0[i] = (byte)(byte2 << 4 & 0xf0 | byte3 >> 2 & 0xf);
            return 2;

        case 3: // '\003'
            abyte0[i++] = (byte)(byte1 << 2 & 0xfc | byte2 >> 4 & 3);
            abyte0[i++] = (byte)(byte2 << 4 & 0xf0 | byte3 >> 2 & 0xf);
            abyte0[i] = (byte)(byte3 << 6 & 0xc0 | byte4 & 0x3f);
            return 3;
        }
        throw new RuntimeException("Internal Errror");
    }


    public static void main(String[] args) {

     String  a="Sangfor";

     byte [] b=null;

     b=a.getBytes();


     String encodeString=B64.encode(b);

      System.out.println(encodeString);

     byte[] decodeByte=B64.decode("3lkHi9iZNK87qw0p6U391t92qlC5rwn5iFqyMFDl1t92qUnL6FQjqln76l-P");


      System.out.println(new String(decodeByte));


    }
}
```

运行得到：



3.Baby_Forenisc

内存取证题，直接上Volatility，先psscan看看进程情况：

```
tom@kali:~/视频/volatility$ ./volatility -f 1.raw --profile=WinXPSP2x86 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID    PPID PDB        Time created                   Tim
e exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ---
------------------------------
0x0000000001c50020 DumpIt.exe         1760   1624 0x02b402e0 2021-09-08 05:30:22 UTC+0000

0x0000000001c606a0 vmtoolsd.exe        488    704 0x02b402a0 2021-09-08 05:16:23 UTC+0000

0x0000000001c63020 svchost.exe         176    704 0x02b40220 2021-09-08 05:16:15 UTC+0000

0x0000000001cb3960 wpabaln.exe        1432    604 0x02b40340 2021-09-08 05:17:57 UTC+0000

0x0000000002132558 csrss.exe           580    512 0x02b40060 2021-09-08 05:15:53 UTC+0000

0x00000000021f6da0 explorer.exe       1624   1552 0x02b40200 2021-09-08 05:15:57 UTC+0000

0x00000000021f77f8 spoolsv.exe        1592    704 0x02b401e0 2021-09-08 05:15:57 UTC+0000

0x00000000022311a0 svchost.exe         884    704 0x02b40100 2021-09-08 05:15:54 UTC+0000

0x000000000230e318 wmiprvse.exe       1444    884 0x02b40300 2021-09-08 05:16:24 UTC+0000

0x0000000002310020 alg.exe            1876    704 0x02b40320 2021-09-08 05:16:25 UTC+0000

0x000000000231fb28 wscntfy.exe        1376   1084 0x02b402c0 2021-09-08 05:16:25 UTC+0000
```


发现有cmd，那就先看看cmd:

```
tom@kali:~/视频/volatility$ ./volatility -f 1.raw --profile=WinXPSP2x86 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: csrss.exe Pid: 580
CommandHistory: 0x565c60 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x524
**************************************************
CommandProcess: csrss.exe Pid: 580
CommandHistory: 0x566bb8 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 5 LastAdded: 4 LastDisplayed: 4
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x4cc
Cmd #0 @ 0x3689ed8: git push -u origin master
Cmd #1 @ 0x566148: ok....
Cmd #2 @ 0x56aa08: then delete .git and flagfile
Cmd #3 @ 0x368a798: You can never find my account
Cmd #4 @ 0x56a580: hahaha!
```


说把flag推到git上了，本地没有。先搜索文件看看：

```
tom@kali:~/视频/volatility$ ./volatility -f 1.raw --profile=WinXPSP2x86 filescan |grep txt
Volatility Foundation Volatility Framework 2.6
0x00000000020bf6a0      1      0 RW-r-- \Device\HarddiskVolume1\Documents and Settings\Owner\桌
面\ssh.txt
0x00000000021c01b0      1      0 R--rwd \Device\HarddiskVolume1\Documents and Settings\All User
s\ssh.txt
0x000000000231d6b0      4      2 -W-rw- \Device\HarddiskVolume1\Documents and Settings\All User
s\Application Data\VMware\VMware VGAuth\logfile.txt.0
```

搜索到了ssh.txt，使用命令`./volatility -f 1.raw --profile=WinXPSP2x86 dumpfiles -Q 0x00000000020bf6a0 -D ./`把ssh.txt提取出来，是一个私钥文件：

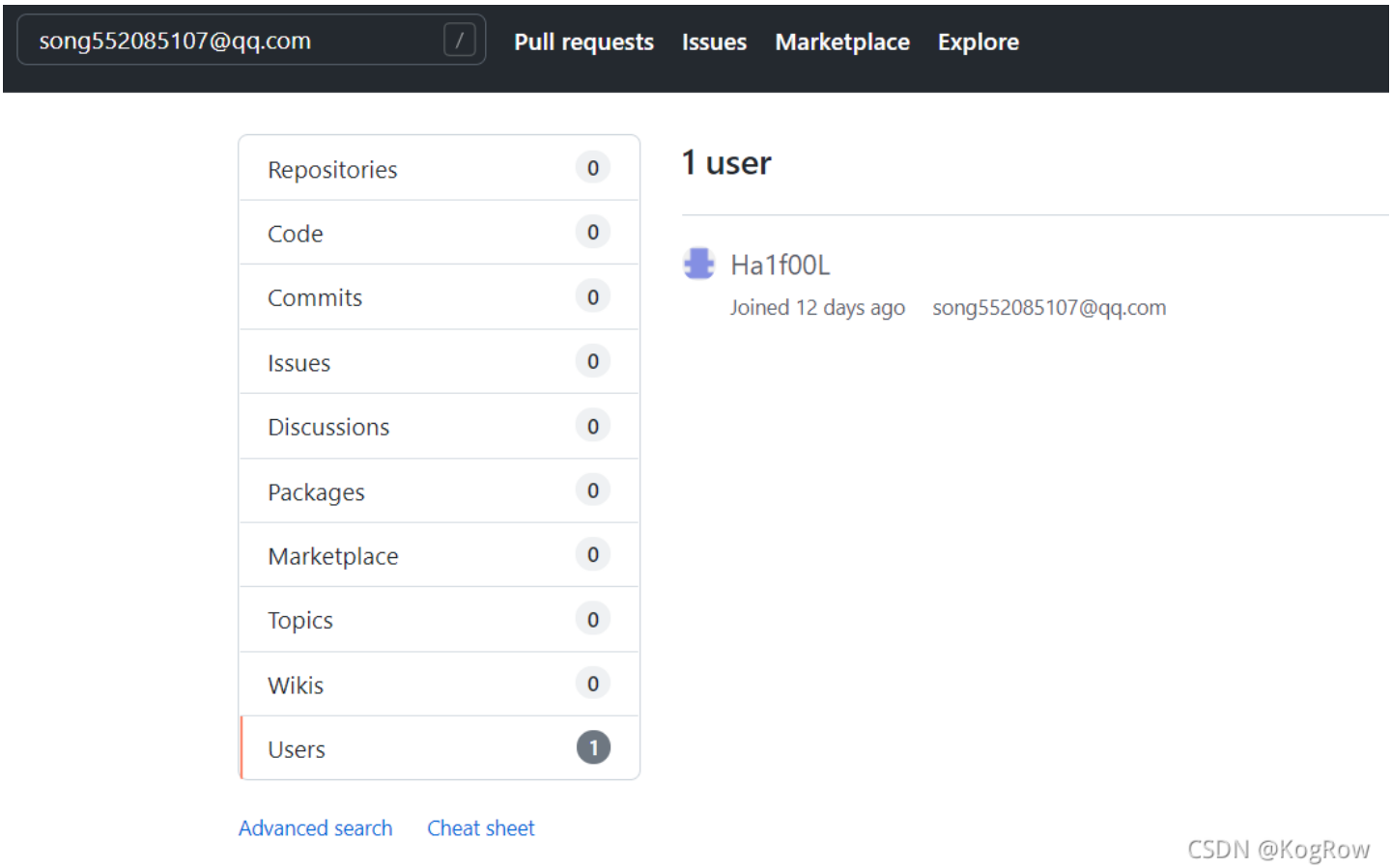
```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAmw8eqi/h23ABuRhhmx83LuRhw6m8C8k76Me0s7MNdvDP2ZB5hJUU
fZ4HxR5sEoQf6NyIcCDeznb8FAYAktm3cBlgof847aL661F0R5FtIfOJC/MwklRmXjYr46
6HNjQ0Ouu12znqBPJAaMkAaZXknqlEAxCRvyOQhg0bPSR3xxCM39TxpXRKd3tzhlBUQHZi
upgt6CF3TkBuIcKUPgZ7OgJ/7ES3FaiUOlpZdUYf/H3VwwQumuXPPwvT5QdRA9Myv/zbee
R9ddLJL84raHK6unuHjngGvWjhXUUQulta49HH55pyrFUViIvH1tfns/6BglTrYWRlFX3A
TNOVy2igHkhZI8M9GK5VUBwEo3kXcWRiK85vAWwmddBd9+c0NERahRg+SNbodsd1JFu0C9
kqJ8/HlOnDfPBsUpD0EY/EbzW5PKbkksp2Vp3z+S0y1aVpX2EJRhq2S5kEEU+V4LLN6uqu
CJzVLeG5Lpnn4V/Ekf/ZpJmmk1Pp9KGFw3tlOqTLAAAFkNMuPgLTLj4CAAAAB3NzaC1yc2
EAAAGBAJsPHqov4dtwAbkYYZsfNy7kYcOpvAvJO+jHtLOzDXbwz9mQeYSVFH2eB8UebBKE
H+jciHAg3s52/BQGAJLZt3AZYKH/OO2i+utRdEeRbSHziQvzMJJUZl42K+OuhzY0NDrrtd
s56gTyQGjJAGmV5J6pRAMQkb8jkIYNGz0kd8cQjN/U8aV0Snd7c4ZQVEB2YrqYLeghd05A
biHClD4GezoCf+xEtxWolDpaWXVGH/x91cMELprlzz8L0+UHUQPTMr/823nkfXXSyS/OK2
hyurp7h454Br1o4V1FELpbWuPRx+eacqxVFYiLx9bX57P+gYJU62FkZRV9wEzTlctooB5I
WSPDPRiuVVAcBKN5F3FkYivObwFsJnXQXffnNDREWoUYPkjW6HbHdSRbtAvZKifPx5Tpw3
zwbFKQ9BGPxG81uTym5JLKdlad8/ktMtWlaV9hCUYatkuZBBFPleCyzerqrgic1S3huS6Z
5+FfxJH/2aSZppNT6fShhcN7ZTqkywAAAAMBAAEAAAGAdfojEsorxpKKPRLX8PbnPb52xD
C46x7Jfmu0iaWKcRz4iEjsrHvhg1JiBxEGmW/992cUSHw6Ck1trq6CcTlF4PzuEVPnNKf0
0ma/WlTD/DkX5Qe7xRqCaNw+uJVqO0utEceWLp7595l6eD+3GJ77u9x96vcIba3ZoKUIPJ
UqrUNibEvRMFoy7oX3eBJWiFWK+P4gr6YG6HsNUJKDyE2WJKUSP+pogwoo/d0Qg7I/VBVK
N39PFnwUG5wcNP5EHezqWQVVln/dltDgOc5IldknTRt4Q3NDrSyNsRpv0EYI2gz+yRu/IE
RR9PHYjH5l6uYwowW34iGi/xloSxG5bDEWOe0eEANCjowiYYrmTLffIQ/AU9w4te/+eWd2
WV56LUuC6k4mEdNhtljMZR/0A+C5EkPzgsTEJEmYLYvqrNejM7Y1UKz3+YZ8m8rT4XcNmf
j5wfJd1TbCu0hB5kZC1DkybYQaMRNnZ3+PjwU2hZBTuh02F787nG5NFkpI96qkWxTBAAAA
wBdaxLNzl/7Dig/neTUAQLa/C1F2cpQt6RcJbzHodgxm8n75a/wdRI4/oCvGJkRgyAnyCE
tgfMnTQ4opmHf5k0U0R/wmCGivcGhg5KIBSSnp9mWt6qclJ8O6vZ5L3rKIgreWzGUDk8IT
W3Lcl5EO0sskpVvp65xncEdv3CefxXVTlkgp4PXgXcxPao633hWA6TAm2zZx7R6fJt0Ex4
x3lVG68ghRE/ZFbF48s8Gy+zRDyA5JEGPWxWddO623IVgG6AAAAMEAyX4CJKSxE5gvJdrw
lhx8dBbVQxw06fPoVlu/z/JTwkPdliuAdp30SV8WbmXUhLvv457WdqAMCwlGs/7xrCW21U
84+VeD9aGM61nSsT7kUzGjdvbjQiHCmys7dwuy/thCrpWFTxI4fjOEYHc3N8S+hBHQRJKk
mEYyBoI3eJ3NhUsGHr1V4LONBKkoUZyC+LjKev06m9qM6R0/0k4cB09pkDVinuFuGk5iDy
YKyjAGiAxFI9ACiZ5NLKTsdaEqtCPfAAAAwQDFAXbSxwbLYWDacBNUm4E7FZsYKkqoIAWQ
3uEQP5Sp7GrCU5dWraGB2wOkX+irMYGDfTk5qG8NLyYoSKVIZwA6ijDliWekL6XdPGJfKK
7xw64Nx6syc7oD7scSzTGNH0m1z+T2rjP3dMDDVhYMHksYcSxikyHNzLR9Z51hCOHeKb1O
8LNW4IrC6AYeXt8sHizSLIagncOuPtSkKiGdR5fn65fHomMzaVQsSJYvwNeSrKXu36NSJm
27AuL6DDE2vJUAAAAUc29uZzU1MjA4NTEwN0BxcS5jb20BAgMEBQYH
-----END OPENSSH PRIVATE KEY-----
```

```
③3÷tÀÃV□□K□q,b□!ÍÌ
è□□^ß,□,Ò,□ □Ã®>Ô¤*!□G□çë□Ç¢c3iT,H□/
¼□□□□□song552085107@qq.com□□□□□□□
```

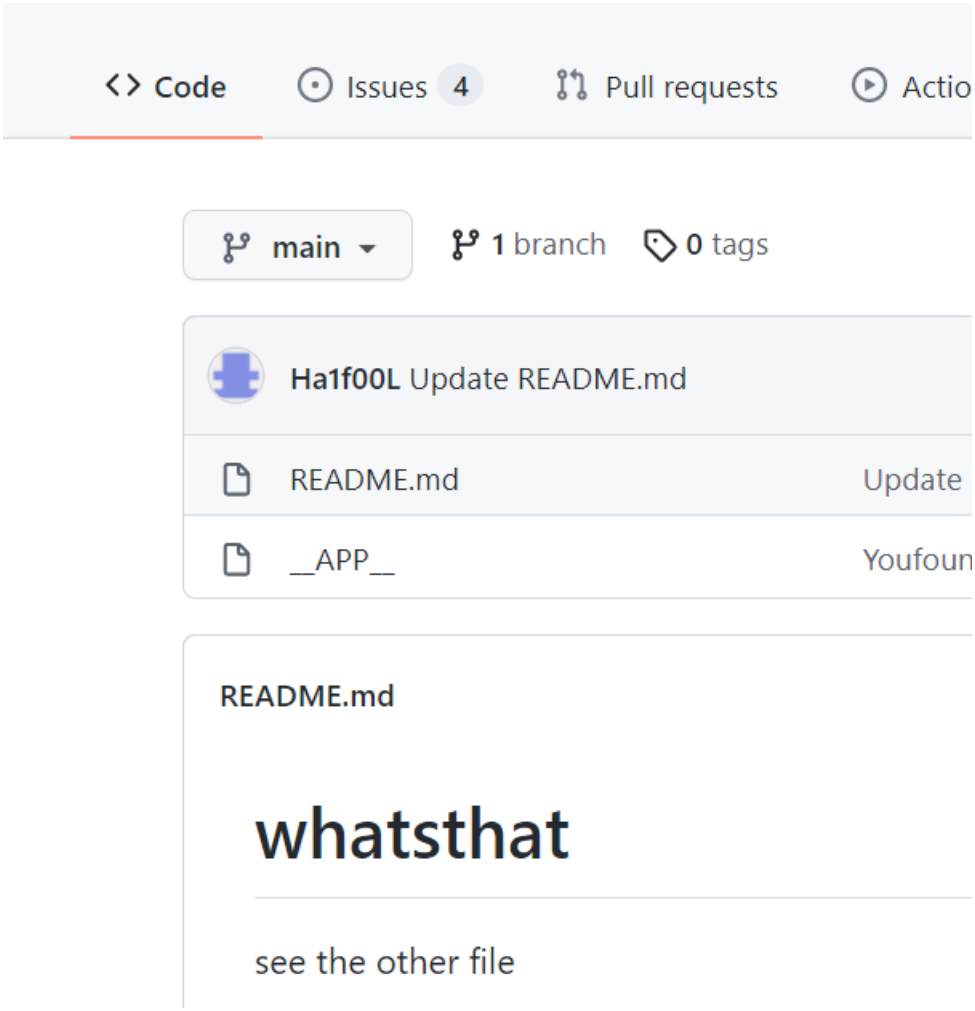
base64解码一下：

看到有一个邮箱，拿去github上看看：

看到有一个邮箱，拿去github上看看：



看看他的提交记录：



只有2个文件，readme没啥，把 __APP__ 下载下来看看：

```
2157 var z=__WXML_GLOBAL__.ops_set.$gwx || [];
2158 __WXML_GLOBAL__.debuginfo_set = __WXML_GLOBAL__.debuginfo_set || {};
2159 var debugInfo=__WXML_GLOBAL__.debuginfo_set.$gwx || [];
2160 function gz$gwx_1(){
2161 if( __WXML_GLOBAL__.ops_cached.$gwx_1)return __WXML_GLOBAL__.ops_cached.$gwx_1
2162 __WXML_GLOBAL__.ops_cached.$gwx_1=[];
2163 (function(z){var a=11;function Z(ops,debugLine){z.push(['11182016',ops,debugLine])}
2164 Z([3,'枴欐榔鏕板滬緇勫欢娴孀癍'],['./pages/index/index.wxml',2,7])
2165 Z([3,'鏕板滬'],['./pages/index/index.wxml',4,7])
2166 Z([3,'枴礬篇鎛  褫璇x95'],['./pages/index/index.wxml',7,7])
2167 Z([3,'浠x80涔堝紝浣犲厛瑕娶\x5c1\x5ca\x5cg?'],['./pages/index/index.wxml',9,7])
2168 Z([3,'U2FuZ0ZvcntTMF8zYXp5XzJfY3JhY2tfbm9vYl9wbGF5ZXJ9'],['./pages/index/index.wxml',10,7])
2169 })(__WXML_GLOBAL__.ops_cached.$gwx_1);return __WXML_GLOBAL__.ops_cached.$gwx_1
2170 }
2171 function gz$gwx_2(){
2172 if( __WXML_GLOBAL__.ops_cached.$gwx_2)return __WXML_GLOBAL__.ops_cached.$gwx_2
2173 __WXML_GLOBAL__.ops_cached.$gwx_2=[];
2174 (function(z){var a=11;function Z(ops,debugLine){z.push(['11182016',ops,debugLine])}
2175 Z([3,'container log-list'],['./pages/logs/logs.wxml',2,13])
2176 Z([3,'log'],['./pages/logs/logs.wxml',3,40])
2177 Z([[7],[3,'logs']],['./pages/logs/logs.wxml',3,17])
2178 Z([3,'log-item'],['./pages/logs/logs.wxml',4,17])
```


2168行，解码：

请将要加密或解密的内容复制到以下区域

SangFor{S0_3azy_2_crack_noob_player}

菜狗落泪。。。。。。。。。

4.Checkin_Go

第一关是md5爆破，直接上python代码：

```python
import hashlib
for a in range(33,126):
    for b in range(33,126):
        for c in range(33, 126):
            for d in range(33, 126):
                for e in range(33, 126):
                    for f in range(33, 126):
                        str = chr(a)+chr(b)+chr(c)+chr(d)+chr(e)+chr(f)
                        fuck = hashlib.md5(str.encode('utf-8')).hexdigest()
                        if fuck[:6] == '2a4d5a':
                            print(str)
                            print(fuck)
                            break
```

登录上去之后就是要想办法变成管理员，增加金钱数购买flag。由于golang语言不是我擅长的，而且没有时间了就没做，等官方wp吧。