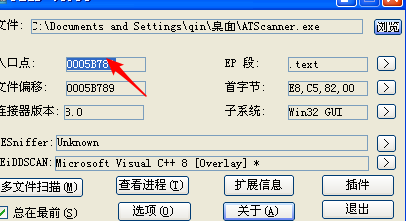
# ATScanner.exe

根据题目提示寻找相关信息即可

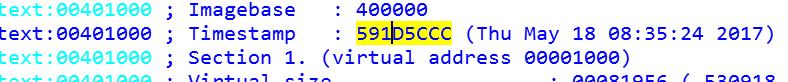
块数量



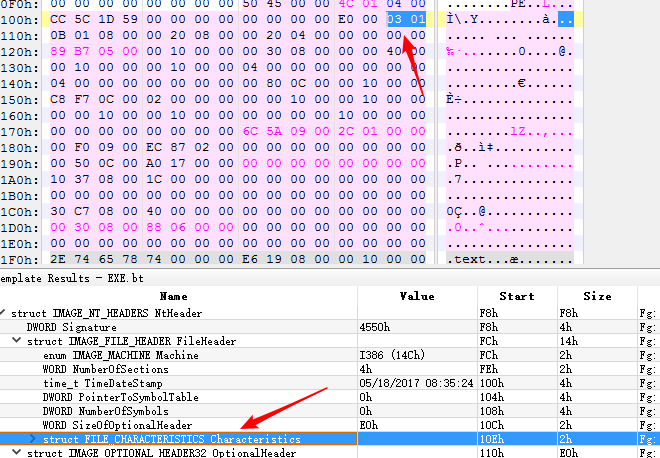
代码入口点 0005B789



时间戳



特征值



所以最终的flag为BJWXB\_CTF{0004h-591D5CCCh-0005B789h-0103h}

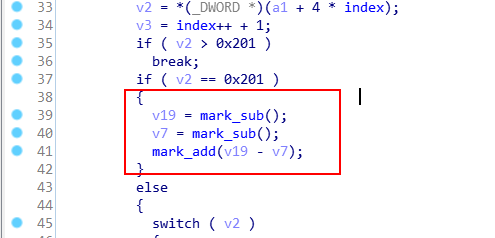
# Warmup

程序会将输入的字符串，每个字符与0xE异或，然后与check字符串” LDYVLQMZHuY:|cQ[^Qyo|cQ{~QYO\CQ[^/s”比较

FLAG: BJWXB\_CTF{W4rm\_UP\_warm\_up\_WARM\_UP!}

# CrackMe

程序像是个解释器，各种复杂的操作，大部分时间都陷进去逆算法了。。。后来发现压根不用搞懂逻辑，程序会将输入的字符串一个字符一个字符和生成的字符比较，判断逻辑如图所示：



v19、v7一个是输入的字符，一个是生成的判断字符，两个相等即可，所以动态调试，通过内存追码即可拿到flag:

BJWXB\_CTF{C5307D46-E70E-4038-B6F9-8C3F698B7C53}

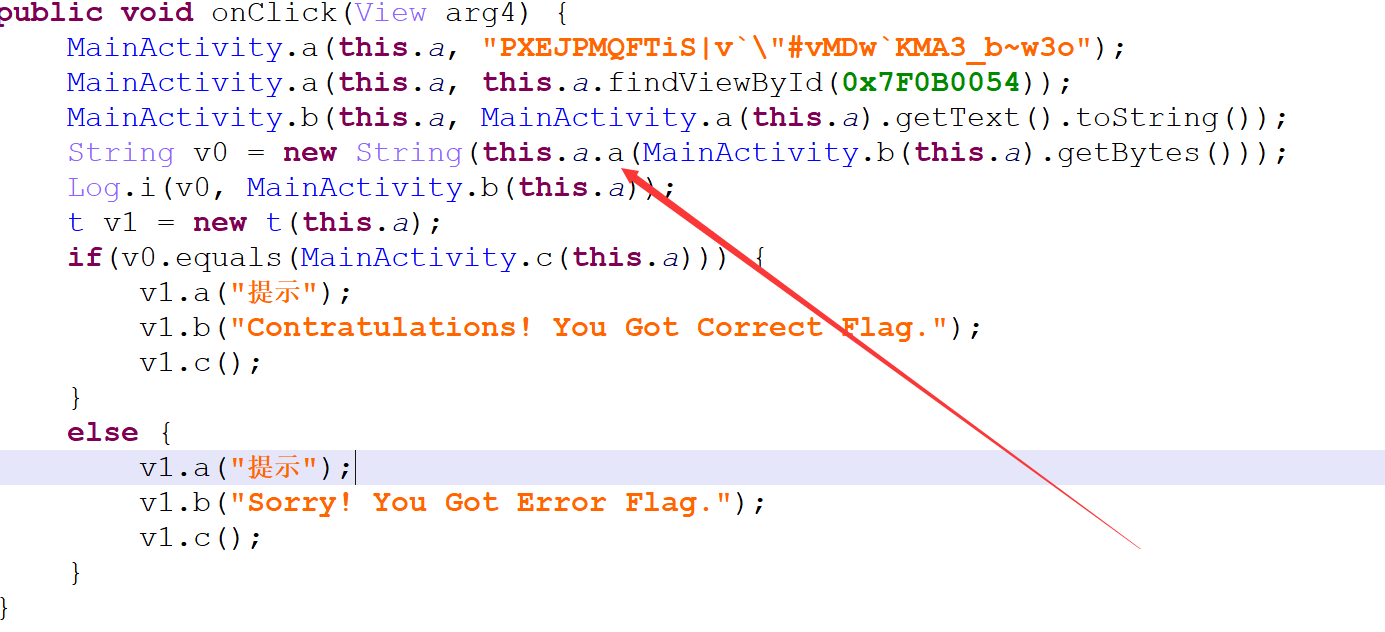
# app-release.apk

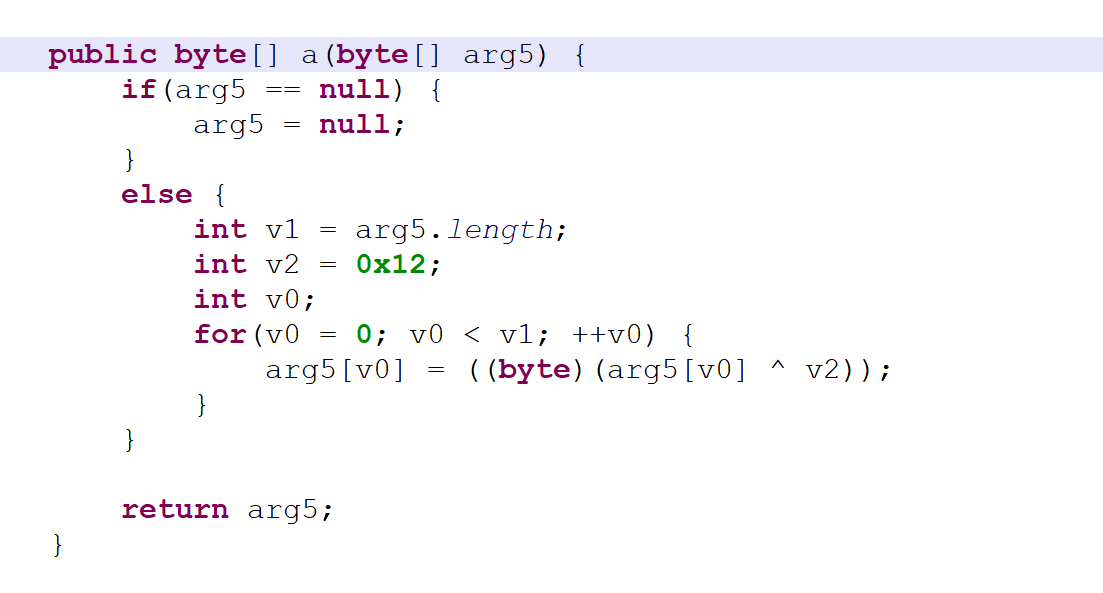
看到有onClickHandler被设置，直接进入查看发现有a.a，双击进入即可看到为简单异或加密。

Python代码

input\_flag = "PXEJPMQFTiS|v`\"#vMDw`KMA3\_b~w3o"

print ''.join([chr(ord(i) ^ 0x12) for i in input\_flag])





# CrackMe.exe

程序加了VMP壳，考虑到时间问题决定带壳调试。发现是mshtml的控件，于是想要dump出来html。经过尝试直接断在CHtmLoad::Write可以成功读取js

console.warn(function (p, a, c, k, e, d) {

e = function (c) {

return (c < a ? "" : e(parseInt(c / a))) + ((c = c % a) > 35 ? String.fromCharCode(c + 29) : c.toString(36))

};

if (!''.replace(/^/, String)) {

while (c--) d[e(c)] = k[c] || e(c);

k = [function (e) {

return d[e]

}];

e = function () {

return '\\w+'

};

c = 1;

}

;

while (c--) if (k[c]) p = p.replace(new RegExp('\\b' + e(c) + '\\b', 'g'), k[c]);

return p;

}('7 6(){a=8.b.9.5;1(a=="4{3-2-h-i}"){0("j!")}k{0("g!<"+a+"> d c f e ;-)")}}', 21, 21, 'alert|if|simpower91|by|BJWXB\_CTF|value|ckpswd|function|document|pswd||all|not|is|GUID|my|wrong|antiy|cn|congratulations|else'.split('|'), 0, {}))

浏览器eval得到

function ckpswd(){a=document.all.pswd.value;if(a=="BJWXB\_CTF{by-simpower91-antiy-cn}"){alert("congratulations!")}else{alert("wrong!<"+a+"> is not my GUID ;-)")}}

找到flag。