

# HW防守 | 基于冰蝎的特征检测

SuPejkj 渗透Xiao白帽 今天

## 前言

临近HW，作为萌新的我在网上找了许多大佬们关于“冰蝎”流量特征的文章，以此作为分享，在我们正式HW时，可能会有所帮助，当监控设备发现这些流量，作为防守方也能做出准确的判断。避免丢分甚至服务器被端🐱

本文内容大多来自网络收集（出处会标在下文）如有问题还望各位大佬指正。

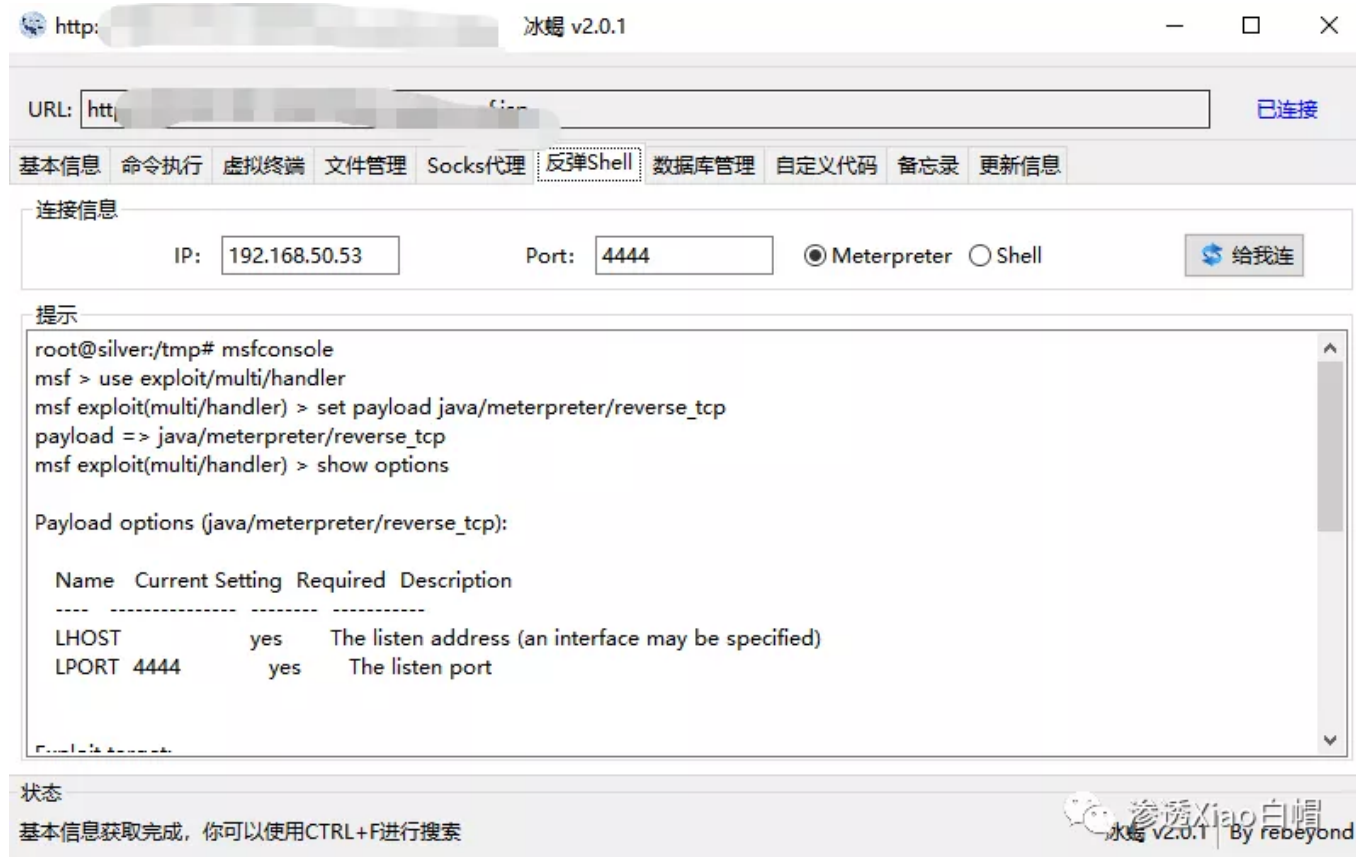
莫要喷俺



## 0x00 简介

“冰蝎”是一款基于Java开发的动态加密通信流量的新型Webshell客户端。

老牌 Webshell 管理神器——中国菜刀的攻击流量特征明显，容易被各类安全设备检测，实际场景中越来越少使用，加密 Webshell 正变得日趋流行。由于通信流量被加密，传统的 WAF、IDS 设备难以检测，给威胁狩猎带来较大挑战。冰蝎其最大特点就是交互流量进行对称加密，且加密密钥是由随机数函数动态生成，因此该客户端的流量几乎无法检测。冰蝎目前最新版本为v2.0.1，兼容性已经日益完善，包括虚拟终端、Socks代理、文件管理、反弹shell、数据库管理等强大的功能，方便部署使用。

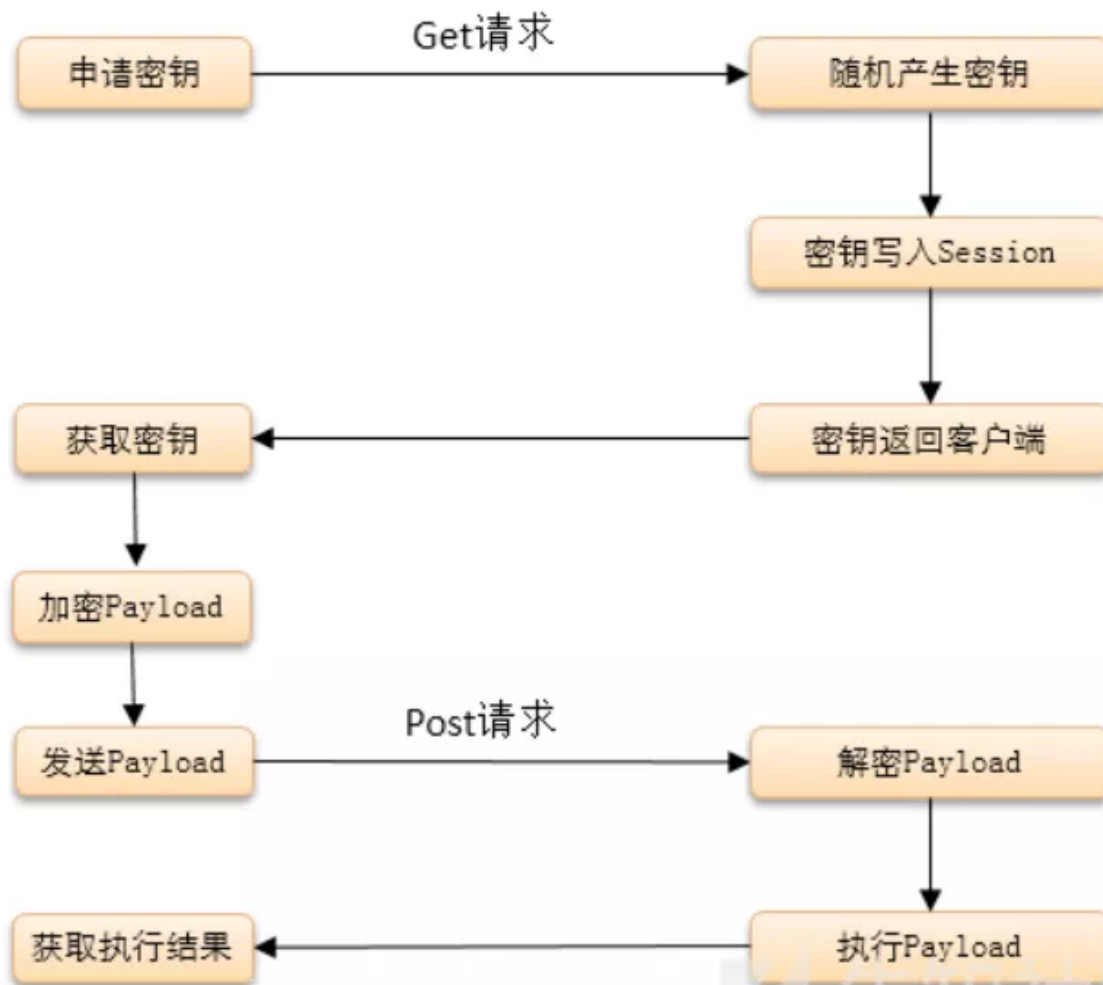


## 0x01 “冰蝎”获取密钥的过程

图片来自红蓝对抗——加密Webshell“冰蝎”攻防

# 客户端

# 服务器



渗透Xiao白帽

冰蝎在连接webshell的时,会对webshell进行两次请求访问

## Request

Raw Params Headers Hex

```
GET /wls-wsat/.conf.jsp?admin@123321=799 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; InfoPath.3)
Host: 192.168.1.100:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: close
Content-Length: 267
```

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 05 Jun 2020 15:16:35 GMT
Content-Length: 16
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: JSESSIONID=ySWFDiFZdW0b20MZz6qID26D-jg7USx3J_odAxH-hf10JFf0ioRK!-1047449674; path=/; HttpOnly
bc4b7c3182d6377b
```

渗透Xiao白帽

# Request

Raw	Params	Headers	Hex
-----	--------	---------	-----

```
GET /wls-wsat/.conf.jsp?admin@123321=33 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; InfoPath.3)
Host: 10.10.10.10
Accept: text/xml, image/gif, image/jpeg, *, q=.2, */*; q=.2
Connection: close
Content-Length: 267

HTTP/1.1 200 OK
Connection: close
Date: Fri, 05 Jun 2020 15:16:42 GMT
Content-Length: 16
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: JSESSIONID=CMGFDJ114oj2Aksf5BVOS0bk2-Ly30E1qLwAetIUEMXiUoGGAYFN!-1047449674; path=/; HttpOnly

964229ad1a4e45cf
```



# Request

Raw	Params	Headers	Hex
-----	--------	---------	-----

```
POST /wls-wsat/.conf.jsp HTTP/1.1
Content-Type: application/octet-stream
Cookie: JSESSIONID=CMGFDJ114oj2Aksf5BVOS0bk2-Ly30E1qLwAetIUEMXiUoGGAYFN!-1047449674; path=/; HttpOnly
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; InfoPath.3)
Cache-Control: no-cache
Pragma: no-cache
Host: 10.10.10.10
Accept: text/html, image/gif, image/jpeg, *, q=.2, */*; q=.2
Content-Length: 8556
Connection: close

Hu23bXudxhGqx766N++ZXIwhPQ7E30fdlgh02P/WLWzRBHskR0fXHjn7d5GJFnEgir5PdF8sBMq1NwqY19qmwmqyMwaTshHKlXbX8hWKt9o4EntN
D417XCWElg696w2u05bCio1f+PbPWIJ1J8c/nNMqYxHxTi6ATyoxBx4kuoqomqg8xdVrP9gdW0e1Gk0AExdXJchZNEs9okkKJqGu37VZCjbVQos
CnzBH/Tb6HYrRLnqRNOtITQWbzWSQ0EqJLVhNEFih0bqiaao7YOGJZe09SUBfXWP40CxoMgu1dkicoUxSCEl6nTENKgv4fN5BWHu/kWlziX0EB
dPPYpV0iqlTdeZT6seJfIYLpnsReH2tG5nj69n+65fbrLEVoE8K14aimrrknbzEsBdCb9x86TZJqVBEv9S1Q+skTxW2yEN4uog3D8Zccad1zJtI
SMZ1XDVTDtT5rSpb81evpftWGFmkk/luHcatJzpH5nVHEgrpj6+H01TY9cpdPD08a6Etf/dwfzME16SC6AonX5Q0CXLGVhm+X81rC687S01xl
+0IRyYvYgY0UGy4deknRBmtDyDpwsjT4pXlGfHkytYe8/1y7MGNW8UoYlI4PBx9+cGKkQNWYXcN7RCteoyFeT+Z3LUT3sxZfpFmY52h1Ysri9F
n8YuIlpmZc0V14p0k/IWHSNGVzQ22FIlPzEPJyIHpxCkxvmdLscUt/s8/a0YYb01EIAZHm8YmZEVEGzw5LPjmlJqsVSw0sxNi94vs6wnbv148af
v9FKHxRlmpU1518+ATKatTG7uz6ew1bV/drWEpOY4CLH0pUDfSUe+5waOYkuTPvu/Y5yqtsBIfZogn1chSxiGHacUpjrUUT9uH+Iq228An5w
wf3SyEgt2MaiEqFP0PHCW25114KdDi3q1PrTr/FldkUzmHvN1k0J6idj1zz/Wn9Uch1cEL2DayvezjjW5S3zBfTRnrpwFqrW5oCX71G52idJdfQ
lpyasU6068ixN8XKJUC4ixKtv4AP3xLSG6iR7ZVHshyWXJJDmhJRIez0L5hb3ZG7XDPWBUJRsmourlb673rm1100XNGWfr/eLeoVJlxybojEhwW
PvNrMjldpJU3xQqaMGyXmchoGcpZKYpoc9e2VMDJHfa3ARGBkZXz1brz662xcUFTottlBjRMD0iqMiRJsYtYfzyN88If0QyVZAX4ziqifELdV0e8
iYJ0b+5DC0+ZBHX1cGVJXNBmnyfInVsb0sMPvGZd5Y2JKrUK13z0pTqVXNOLL98LbXkYbfXu0e46HqFadssDjtMj0vbE5YfruVqXWdiWmbJyINQ
RRzCLkKwdXDXNGWfr/eLeoVJlxybojEh6t8uzniBKNZbT8PnW0Jl0iozwLDJ1M0wJJSUS1MyrXqk3VAgHVuBwzcZff8pvmIrrqG5eNtmeDCFEw
NNpGWE0Y8161Z+uLeJNuE/Jfhd3Dp3c348caS34yYzwna11vm4XPWU1SuhQ9cUxw3Nm2XC4VwJpb6/qZcXQicRoSmWfQroJE0Eq5pcKw4i
H14fMjXgS3n2B+RUFKA3nBDJG0vb41Zj00UjGxT0tyrQjoFwX03Sje2Sc05h7Yw1G17LivyvRAqMYUvMi7fHDUIZGK6HqsxbAUpA0AV1xyNe7Q
L01sCfxZAWABYhwhJhyGw31UjS0YQjUhmZQtRKQRTiFYq85ftFZswy70bpKHSHN/nSaFWe4GEosvTiYU/CyQ3W5PS+GQyLyjshXEdGgvgchqf
Wc423ifS0i/4eC7ScZaVcxV2ABu6hctxS9oo71QB4WIE2uzjScRhgVtn9kvJbghdiEm2KHSBeItn81MzBoQr5eqUG3KxVzzUmntv3v5H90/fpRH
Z2tiYfAmMGZJK5HCKGvU2NTNj6aHrmzhpdo5IjZphJIXNcEaAGxxs+sAZQo5xUnM429kHeF+/JG1axuh1qJ04wtVtTocU0gkpyVzc0ey7U2pRs
12tBmzPpyAC4STAeKUh7zL2XjP3Ct0gPog+JQHiv59Sp50D4+6iJrEcL5GaQj+G+2W18aA+V4mJlVd/rGi1lbhXf8M3AascyzFXf06gqenWslG98y
fFvzFTnagLcvctohVx5aVkh70ytrR1M1cUs60A8Guyp6+ExndlrNX/3zSLf1qN0Jze9sNUSBP09H58+Fmk2eM7wT4PwW2t12Szz9xmJzclR/lXq
Zng+X4NLhu5e0wPLhgdbLZirGGWRfbr66cnjIDSSZ6uieEQLM3xHV8qoEA+eP1tdbn0bvjLASuTanTKzSwgPvOdLHbUGA9cgESGx1TaWc9CK0
T8bm7vcXiUD+8icr/R/VthZHL8Kk2cIXhAGvniTh8AG0w6k/kt1P1OS/LBXpB8XKPMERx2yZtUbl1zT01J3v07361Qayjkfy12VoW0tQW10NtApJc
tNp8W5s0TmbpmTPWYnGkQ0V21av57S3+Xbpx4D08KIPMaMq+Z7NwVQ0Q2jYF4LsBlbKeiX2c5B2+C1c1r+9MorFB83/K460mxjwldQaioQk
a3045jEinDVZUQFkdjmcchdirAvn9j+GfpVUvLWEVFLPvtneXaL+T+L7tLd1tCi19tzbXBHJc10P0ZfjTkyftlBk/30cZ131jYkqtQqXfMK10pV
... (truncated) ...
```

# Response

Raw	Headers	Hex	Render
-----	---------	-----	--------

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 05 Jun 2020 15:28:27 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 22832

|00#00000-00v0000q04z1b00-00
000%00|V00:00[V04U000=000000
000{000T0N000{0[90sr002000dp

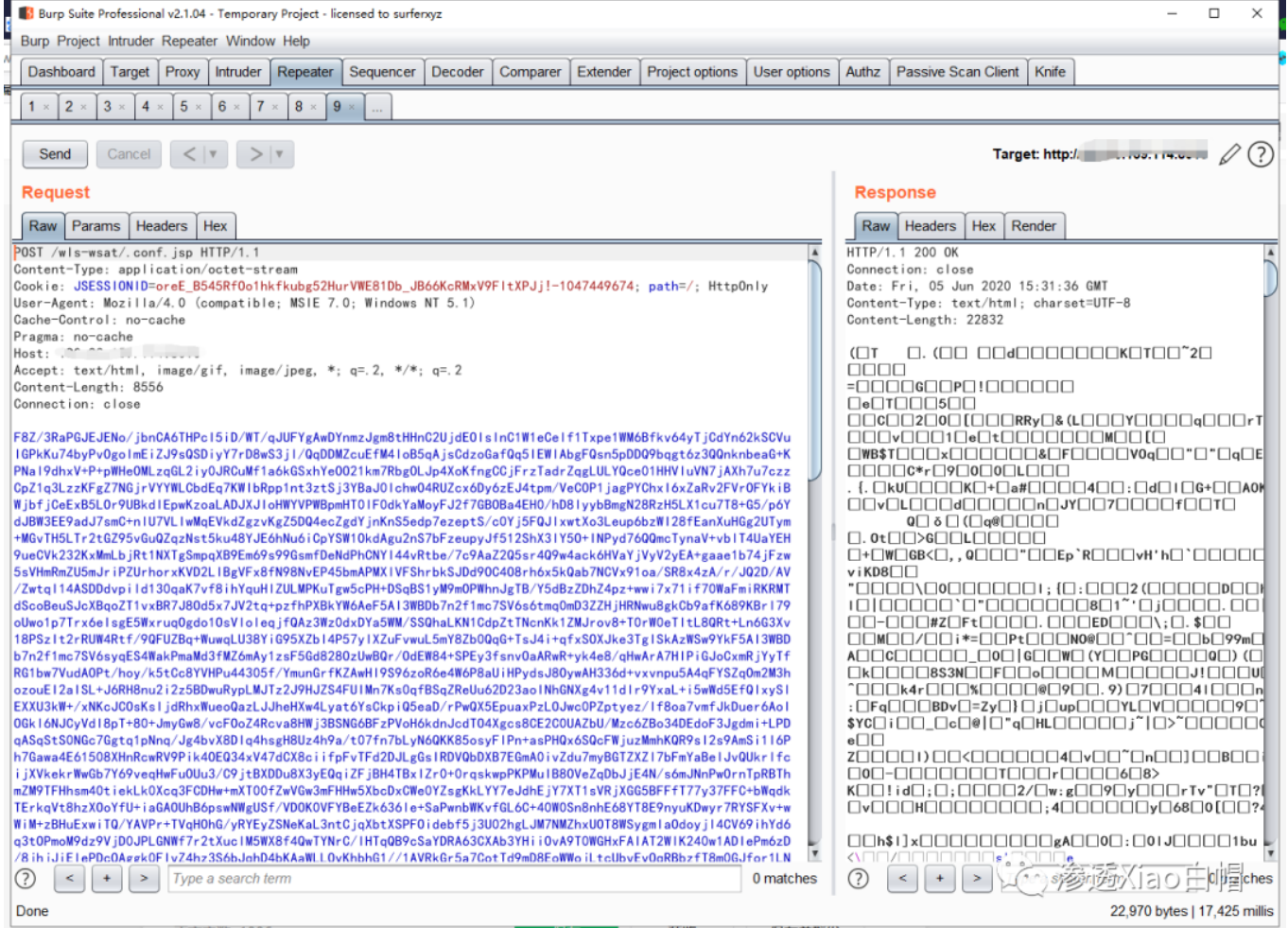
000m;7060\00*0000~<h0;0>00b6q00
000A`000d0A000000;+008r00**00
0;&0t,00000;0XQ:000000~Y0$[090
0b0$00900200i0:,91b00#00000_00
0000000000000000
h$1o0>0|fm3003000,00000000S0
00000(WN000M0ih00000)00Y00f0a0c
uA100+V%000000a1i00K0000`000000
0I0)000000u0000000/00~z000
0V0`10V00000v00000X000f00z00
M00000000
0000XnJt;]04a0000c0000'600000
;g0JD000A00000o00F00'L000TC00
0000r
0{V0X00c000z000=w\W [

0000000`D0u0u0v\40hs000Va000c
00k00;0000000000Bx000000$

00R00000M0V000E000000000000000
00V)p0*00j100(
n00f007000.0000m000w0N0.000

#w3.s0000m00000000000000000000
00000000008*m0000f000;090700
EOjL000n0B;00.00m0V000080000
000000000000000000000000000000
```





为什么进行两次访问?在别的文章没有看到关于这个问题的答案,于是去反编译冰蝎源码

```
180 if (error) {
181     throw new Exception(errorMessage);
182 }
183 String rawKey_1 = sb.toString();
184 if (!Pattern.compile("[a-fA-F0-9]{16}").matcher(rawKey_1).find()) {
185     throw new Exception("页面存在, 但是无法获取密钥!");
186 }
187
188 int start = 0;
189 int end = 0;
190 int cycleCount = 0;
191 while (true) {
192     Map<String, String> KeyAndCookie = getRawKey(getUrl(), password, requestHeaders);
193     String rawKey_2 = KeyAndCookie.get("key");
194     byte[] temp = CipherUtils.bytesXor(rawKey_1.getBytes(), rawKey_2.getBytes());
195     int i = 0;
196     while (true) {
197         if (i >= temp.length) {
198             break;
199         } else if (temp[i] <= 0) {
200             i++;
201         } else if (start == 0 || i <= start) {
202             start = i;
203         }
204     }
205     int i2 = temp.length - 1;
206     while (true) {
207         if (i2 < 0) {
208             break;
209         } else if (temp[i2] <= 0) {
210             i2--;
211         } else if (i2 >= end) {
212             end = i2 + 1;
213         }
214     }
215     if (end - start == 16) {
216         hashMap.put("cookie", KeyAndCookie.get("cookie"));
217         hashMap.put("beginIndex", new StringBuilder(String.valueOf(start)).toString());
218         hashMap.put("endIndex", new StringBuilder(String.valueOf(temp.length - end)).toString());
219         hashMap.put("key", new String(Arrays.copyOfRange(rawKey_2.getBytes(), start, end)));
220         return hashMap;
221     } else if (cycleCount > 10) {
222         throw new Exception("Can't figure out the key!");
223     } else {
224         cycleCount++;
225     }
226 }
```

HACK学习呀



```

362 public static Map<String, Object> requestAndParse(String urlPath, Map<String, String> header, byte[] data, int beginIndex, int endIndex) throws Exception {
363     Map<String, Object> resultObj = sendPostRequestBinary(urlPath, header, data);
364     byte[] resData = (byte[]) resultObj.get("data");
365     if (!beginIndex == 0 && endIndex == 0 && resData.length - endIndex >= beginIndex) {
366         resData = Arrays.copyOfRange(resData, beginIndex, resData.length - endIndex);
367     }
368     resultObj.put("data", resData);
369     return resultObj;
370 }

```

先知社区

```

Server: Microsoft-IIS/7.5
X-Powered-By: PHP/5.5.38
Set-Cookie: PHPSESSID=s3n3is01p1kb34sc5ffa5cshb3; path=/
X-Powered-By: ASP.NET
Date: Tue, 14 Apr 2020 13:17:18 GMT
Content-Length: 25

gif89aa
976b3dae2f7bfc48

```

渗透Xiao白帽

通过对代码阅读,可以发现冰蝎为了实现可以在webshell内添加任意内容 (比如gif89a子类的文件头或者其它标示字符) 冰蝎在初始化密钥时会两次对webshell进行访问,然后比较两次页面返回的差异,把两次请求都相同的字符记录一个位置,后续加密会用到这两个位置(beginIndex,endIndex)

\*new 1 - Notepad++

文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗



new 1 x

```

1 gif89aa
2 976b3dae2f7bfc48
3 asd

```

第一次请求

```

5 gif89aa
6 70005670eb029dbf
7 asd

```

第二次请求

HACK学习呀

new 1 x

```

1 gif89aa
2 976b3dae2f7bfc48
3 asd
4
5 gif89aa
6 70005670eb029dbf
7 asd
8
9 gif89aa
10 我是被加密的数据
11 asd

```

HACK学习呀

```

if (!beginIndex == 0 && endIndex == 0 && resData.length - endIndex >= beginIndex) {
    resData = Arrays.copyOfRange(resData, beginIndex, resData.length - endIndex);
}
resultObj.put("data", resData);

```

HACK学习呀

先知社区

```
try {
    response.setContentType("text/html");
    pageContext = _jspxFactory.getPageContext(this, request, response,
        null, true, 8192, true);
    _jspx_page_context = pageContext;
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    _jspx_out = out;
```



)

如图,根据数据包,beginIndex:8 endIndex:4 (含换行),冰蝎开始从数据流中截取被加密的数据从下标8开始到(数据包总长度-4)

Waf可以针对于返回类型为 "text/html" 的数据包中加一些空格或者换行,来扰乱冰蝎的数据包,导致冰蝎无法运行。

(为什么要对返回类型为 "text/html" 的扰乱,别的格式不可以吗?)

答案:jsp默认返回类型就是 "text/html" html添加一些空格或者换行,并不会影响网页的正常运行)

## 0x02 "冰蝎" 解析Cookie流程

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Content
9	http://...	GET	/wls-wsat/ conf.jsp?admin@123321=799	✓		200	263	text	jsp		
10	http://...	GET	/wls-wsat/ conf.jsp?admin@123321=33	✓		200	263	text	jsp		
11	http://...	POST	/wls-wsat/ conf.jsp	✓		200	22970	HTML	jsp		
12	http://...	POST	/wls-wsat/ conf.jsp	✓		200	22970	HTML	jsp		

Request

Response

Raw

Headers

Hex

Render

```
HTTP/1.1 200 OK
Connection: close
Date: Fri, 05 Jun 2020 15:16:42 GMT
Content-Length: 16
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: JSESSIONID=CMGFDj1I4oJ2Aksf5BVOS0bk2-Ly30E1qLwAEtIUeMXiUoGGayFN!-1047449674; path=/; HttpOnly

964229ad1a4e45cf
```



Filter: Hiding CSS, image and general binary content											
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Co
9	http	GET	/wls-wsat/.conf.jsp?admin@123321=799	✓		200	263	text	jsp		
10	http	GET	/wls-wsat/.conf.jsp?admin@123321=33	✓		200	263	text	jsp		
11	http	POST	/wls-wsat/.conf.jsp	✓		200	22970	HTML	jsp		
12	http	POST	/wls-wsat/.conf.jsp	✓		200	22970	HTML	jsp		

Request	Response
Raw	Params
Headers	Hex

```

POST /wls-wsat/.conf.jsp HTTP/1.1
Content-Type: application/octet-stream
Cookie: JSESSIONID=CMGFDj114o2Aksf5BVS0Sbk2-Ly30E1qLwAetIUeMxiUoGGAYFN!-1047449674; path=/; HttpOnly
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; InfoPath.3)
Cache-Control: no-cache
Pragma: no-cache
Host: 192.168.1.100:8080
Accept: text/html, image/gif, image/jpeg, */*; q=.2, */*; q=.2
Content-Length: 8556
Connection: close

Hu23bXudxhGqx766n++ZXlwhPQ7E30fdIgh02P/WLWzR8HSkR0fXHjn7d5GJfNfEgir5PdF8sBMq1NwqY19mqwmyMwATShHKIXbX8hWKT9o4EntND417XCWeIlg696w2u05bC1oIf+PBpWlJlJ8c/nNMqYxHxTi6ATyoxB
x4kuoqmqg8xdVrP9gDWOe1GkQAEadXJchZNEs9okkjqGuJ37VZCjbVQosCnzBH/tB6HYRLnqRNOtITQWBzWSQ0EqJLVhNEfIhObqiaao7YOGjZeQ9SUBfxWP40CXMguIdKicoUXSCEl6n7ENKgv4fN5BWHu/kWlzi
XOEBPPYV0iqlTdeZT6sjeF1YLpnsReH2tG5nj69n+65fbrLEvE8K14aimrrknzbEsBdCb9x86TJqVBEv9SIQ+skTxW2yEN4uog3D8ZcadIzJtISMZ1XDVTDT5rSpb81evpftWGFmkk/luHcatJzph5NVHEgrpji
6+H01TY9cpDD8a6Etf/dwfzME16SC6AonX5Q0CXLcVhm+x8lrC687S0lx+0iRyYvFyYUGy4deknRBmMt0yDPwsJ4pXlGfHkytYe8/1y7MCW8UoYlI4PBx9+cGKkQNWYXcN7RCteoyFeT+Z3LUT3szZfpFmY52h
1YsrI9Fn8YUlpMZcOV14p0k/IWhSNGVzQ22FLpZEPJyIHpxCkxvmdLscUt/s8/a0YYb01EAZHMBYmZEVeGzw5LPjmWjqsVSw0sNik4vs6wnbvI4Bafv9FKHXLrmmpU1518+ATKatTG7uz6ew1bV/drWEp0Y4CLqH0
pvUDfSU+5wa0YkuTPvu/Y5yqt8iFZogn1chSxiGhaCUpjRUT9uH+Iq228An5wff3SyEgt2MaiEqFOPPHCWZ5114kCdI3q1PrTr/FIdkUzmHvnlk0J6idj1zz/Wn9Uch1cEL2DayvezjjUWS3zBfTRnrpwFqrW5oCX7
IG52iDjdfQlpyasU6068ixN8XKUC4lxKtv4AP3xLSGq6iR7ZVhshyWXJJDmhJR1ez0L5hb3ZG7XDPWBUJRsmour1b673rml100XNGWfr/eLeoVJlxybojEhwWPvNrMjLdpJU3xQqaMGyXmchoGcpZkYPoc9e2VNDJHfq3
ARGbkZx1brz662xcUFT0tt1bJRMD0iqMiRJsYTFzYn88lfoQyVZAX4ziqfELVQeBiYJOb+5DC0+zBHX1cGVJXN8myfInVsb0sMPvG2d5Y2JKrUK13zCpTqVXNOLL98L1YbFXu0e46HqFadsdJtMj0ybE5YfrvU
qXWdWmbJyINGRRzOLkKwDXNGWfr/eLeoVJlxybojEh6T8uzniBKNZbT8PnW0JJoiozWLDJIMoawJSUS1MyrXk3VAgHVuBWzcZff8pvmIrrqC5eNTmeDCFEwNpGWE1277vYxKqMYUvW17FH0U2Gk0nqSx
34yyZwna1vm4XPWJ1Suh09cUXw3Nm2XC4VwJpb6/qZcX0icRoSmWFQroJE0Eq5pcKw4iH14fMJxgS3n2B+RUFKA3nBDJg0vb41Zjc00UjGxT0tyrQjofWx03Se2Se05h7Yh17LTVyKqMYUvW17FH0U2Gk0nqSx
baUpA0AVIxyNe7QL01sCfxZAWABYhwHjhyGW31UJSQYCUhMz0tRKQRTIFyq85ftFZswy70bpKHSN/nSaWe4GEosvTiYU/CyQ3W5PS+GQyLyjshXEdGgvgchqFWc423ifS01/4eC7ScZaVcxVZABu6hctxS9oo7I
QB4WIE2uzjScRqhVtw9kvJbghdiEm2KHSBelm81MzBoQr5eqJG3KvZzUmntw3v5H90/fpRHvztYfAmMGzJkK5HCKGvU2NTNJ6aHrzmzhzpdo5IzphJIXNcEAGxxs+sAQzo5XnM429kHeFrJG1axuhiqJQ4wtTO

```

```

Iterator<String> it = headers.keySet().iterator();
while (true) {
    if (!it.hasNext()) {
        break;
    }
    String headerName2 = it.next();
    if (headerName2 != null && headerName2.equalsIgnoreCase("Set-Cookie")) {
        for (String cookieValue : headers.get(headerName2)) {
            cookieValues = String.valueOf(cookieValues) + ";" + cookieValue;
        }
        if (cookieValues.startsWith(";")) {
            cookieValues = cookieValues.replaceFirst(";", "");
        }
    }
}

```

HACK学习呀

先知社区

我们可以看到请求协议头中的Cookie字段,冰蝎在合并处理Cookie的时候没有考虑到,Cookie的一些属性 (比如 Path 或者 HttpOnly 之类或者其它的) 冰蝎直接把返回协议头中的Set-Cookie字段直接添加到下一个请求包的Cookie字段中

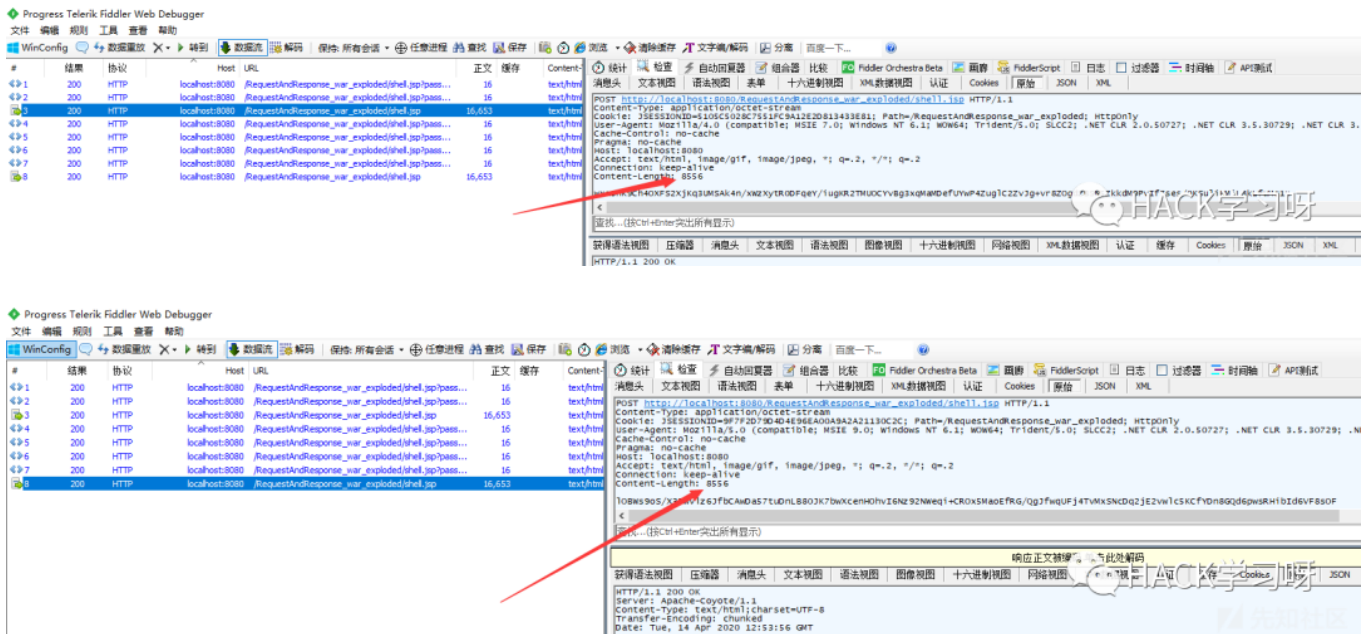
The screenshot shows a Fiddler Web Debugger interface. The top pane displays the request and response details. The request headers include a cookie: JSESSIONID=BA981674A2890B6A45EFDF014F492E; Path=/; HttpOnly. The response headers include a Set-Cookie: JSESSIONID=BA981674A2890B6A45EFDF014F492E; Path=/; HttpOnly. Red arrows point from the response Set-Cookie to the request cookie field, indicating that the response cookie is being added to the request cookie.

HACK学习呀

先知社区

正常的请求是不会携带Cookie属性的,这可是识别冰蝎流量最直接的一种办法

## 0X03 "冰蝎" 动态加载



冰蝎动态加载的原理就是每次都发送一个class字节码(其它语言也一样) 冰蝎通过asm动态修改class字节码变量内容,实现携带参数动态执行,冰蝎在获取完密钥之后(2个请求),第三个请求就是获取BasicInfo(服务器的一些信息),冰蝎的BasicInfo功能并没有动态修改参数(一个获取服务器信息的能有啥参数),这会导致每次获取BasicInfo的数据包都是固定的大小。

## 0x04 UserAgent字段（可绕过）

冰蝎内置了十余种UserAgent，每次连接shell会随机选择一个进行使用。

以下UserAgent列表是从冰蝎的jar包中提取的，可见大多是比较早的浏览器，现在很少有人使用。而且有些国产浏览器甚至精确到了小版本，众所周知，很多国产浏览器是默认自动更新，正常用户很少用过早的版本，因此可以作为强特征使用。

如果发现历史流量中同一个源IP访问某个URL时，命中了以下列表中多个UserAgent，那基本确认就是冰蝎了。



Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1(KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50(KHTML, like Gecko) Version/5.1 Safari/534.50 " BOpera/9.80 (Windows NT6.1; U; zh-cn) Presto/2.9.168 Version/11.5

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; .NET4.0C; InfoPath.3)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;Trident/4.0; GTB7.0)

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1) , 7

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Mozilla/5.0 (Windows; U; Windows NT 6.1; ) AppleWebKit/534.12 (KHTML, like Gecko) Maxthon/3.0 Safari/534.12

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)

Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.3 (KHTML, like Gecko) Chrome/6.0.472.33 Safari/534.3 SE 2.XMetaSr

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)

Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML,like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.20

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E) QQBrowser/6.9.11079

Mozilla/5.0 (compatible; MSIE 9.0; WindowsNT 6.1; WOW64; Trident/5.0)



Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.1(KHTML, like Gecko) Chrome/14.0.835.163 Safari/535.1

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:6.0) Gecko/20100101 Firefox/6.0

Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.50(KHTML, like Gecko) Version/5.1 Safari/534.50 " BOpera/9.80 (Windows NT6.1; U; zh-cn) Presto/2.9.168 Version/11.5

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64;x64; Trident/5.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; Tablet PC 2.0; .NET4.0E)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; .NET4.0C; InfoPath.3)

Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;Trident/4.0; GTB7.0)

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1) , 7

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)

Mozilla/5.0 (Windows; U; Windows NT 6.1; ) AppleWebKit/534.12 (KHTML, like Gecko) Maxthon/3.0 Safari/534.12

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E; SE 2.X MetaSr 1.0)

Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.3 (KHTML, like

Gecko) Chrome/6.0.472.33 Safari/534.3 SE 2.XMetaSr

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2;.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0;InfoPath.3; .NET4.0C; .NET4.0E)

Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML,like Gecko) Chrome/13.0.782.41 Safari/535.1 QQBrowser/6.9.11079.20

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;Trident/5.0; SLCC2;.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;Media Center PC 6.0;InfoPath.3; .NET4.0C; .NET4.0E) QQBrowser/6.9.11079

Mozilla/5.0 (compatible; MSIE 9.0; WindowsNT 6.1; WOW64; Trident/5.0)

*同样，UserAgent可由黑客自定义，因此该特征可能会被绕过。*

## 0x05 总结

WAF可以对一个ip连续访问2次的数据包进行截取,比对相同字符,比对之后,截取两次不同的数据,如果剩下的是16位的key,就可以证明这两个数据包就是冰蝎发出的,第三个数据包通过**“冰蝎”解析cookie流程**和**“冰蝎”动态加载**中的一些bug,可以100%的匹配到冰蝎流量,不会误报。

## 0x06 参考链接

- 1 <https://mp.weixin.qq.com/s/ZD34UQ0gP5cvDctVoIatIA> // HACK学习呀 BeichenDream
- 2 <https://www.freebuf.com/articles/web/216133.html>
- 3 <https://zhuanlan.zhihu.com/p/135227454>
- 4 <https://blog.csdn.net/dianzhongsou2379/article/details/100599116>
- 5 <https://www.cnblogs.com/guojia000/p/11641023.html>
- 6 [https://blog.csdn.net/qz\\_36334464/article/details/99978193](https://blog.csdn.net/qz_36334464/article/details/99978193)

**这里推荐一位大佬的文章聚合平台**

所有资料应有尽有👉

- 1 <http://wechat.doonsec.com>