

Trojan.Congur 事件說明

1

林宜進

簡介

2

- TACERT 陳老師在12/12日信中問有關於Congur資安事件相關問題，為以下這三點：
 1. 此類資安事件的偵測規則之特徵內容為何？
 2. 是APP本身權限問題嗎？
 3. 有其他偵測的特徵嗎?(信中舉的範例是IP位置)

IPS上的規則

3

- 內對外連線 (dst port 是 6280)
- 封包內容包含以下特徵(共五個)
 1. /collect
 2. ts=
 3. ver=
 4. diff=
 5. hash=

廠商的確認回覆

4

Dear 宜進,

1. IP問題，從IPS上看，這些觸發事件的目標IP會連到韓國和大陸的IP，這個事件的規則是否和目標IP無關?

此事件規則與目標IP無關

2. 封包特徵，從IPS上看到的規則是要符合5個特徵才會觸發，請問這些特徵的內容是甚麼?

依照snort rule，只要符合下列5個特徵即會觸發告警

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 6280 (msg:"MALWARE-CNC Andr.Trojan.Congur variant outbound connection detected"; flow:to_server,established; content:"/collect"; fast_pattern:only; content:"ts="; nocase; content:"ver="; nocase; content:"diff="; nocase; content:"hash="; nocase; metadata:impact_flag red, policy balanced-ips drop, policy security-ips drop, service http; reference:url,virustotal.com/en/file/cd050a2868646f231c544c566ba3cc34538e63a5125a2c7ad1f4bbd41d5e8cdd/analysis/; classtype:trojan-activity; sid:44554; rev:1; )
```

```
POST /collect?ts=1459318632000&ver=25483557&diff=1989&hash=00850463526 HTTP/1.1
Content-Length: 480
Content-Type: application/octet-stream
Charset: UTF-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; Redmi Note 3 MIUI/V8.1.1.0.MHOMIDI)
Host: 123.56.205.151:6280
Connection: Keep-Alive
Accept-Encoding: gzip

...i. .3..M69N{..x..R.._.....#..C..n:... ..a.....-.b>.-..>(..$.)[...
9.!.....:0.6.^;.,|,.Qe.V...N...E.:YL.....A.....-ef...If....f.!G...C3o.6..B.}'. 7_
$.#..D.....y..%.0)...3Y..P..<#...~.B...D.....8..w....^[...>cT.\.. -.z}Hi..).....)2.
:t.....,|d?.gr.....7....Z.
1..u..)0A.0e.n3.-..xR{^P..N.&..R]1..q.Y.f.n.l..&F).S/..FJ.XD...
#...S..q738.|2..Q.M...EKD..Z.....F.....u.....R*.....b...&T..
2.h...F4."P.tS..x..h.^..o..R.R.Mf...\\1.k.F....~...m...9@.$
.@|.%.!!!S....<$_.?
```

IPS上的封包範例-1 (dst port 6280)

5

The screenshot shows a Wireshark interface with a single packet selected. The packet list pane shows:

No.	Time	Source	src port	Destination	dst port	Protocol	Length	Info
1	0.000000	140.129.7.154	56339	1.201.143.134	6280	HTTP	837	POST /collect?ts=1511403991000&ver=210578&diff=125&hash=06870236239 HTTP/1.1 (application/octet-stream)

The packet details pane shows the following structure:

- Frame 1: 837 bytes on wire (6696 bits), 837 bytes captured (6696 bits)
- Ethernet II, Src: Cisco_30:7d:32 (78:e4:22:30:7d:32), Dst: Cisco_31:d3:51 (78:e4:22:31:d3:51)
- 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 101
- Internet Protocol Version 4, Src: 140.129.7.154, Dst: 1.201.143.134
- Transmission Control Protocol, Src Port: 56339, Dst Port: 6280, Seq: 1, Ack: 1, Len: 779

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 78 e4 22 31 d3 51 78 e4 22 30 7d 32 81 00 00 65  p."1(p. "0}2...e
0010 08 00 45 00 03 33 2e 39 40 00 7b 06 a9 21 8c 81  ..E..3.9@.{...
0020 07 9a 01 c9 8f 86 dc 13 18 88 0d bb 99 92 db 26  .....
0030 2c 2c 50 18 01 00 0c b2 00 00 50 4f 53 54 20 2f  ,,P.....:POST /
0040 63 6f 6c 6c 65 63 74 3f 74 73 3d 31 35 31 31 34  collect? ts=15114
0050 30 33 39 39 31 30 30 30 26 76 65 72 3d 32 31 30  03991000 &ver=210
0060 35 37 38 26 64 69 66 66 3d 31 32 35 26 68 61 73  578&diff=125&has
```

IPS上的封包範例-1 (符合特徵)

6

Wireshark · Follow TCP Stream (tcp.stream eq 0) · packet

1. 2. 3. 4. 5.

POST /collect?ts=1511403991000&ver=210578&diff=125&hash=00870236239 HTTP/1.1
Content-Length: 480
Content-Type: application/octet-stream
Charset: UTF-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; m2 Build/LMY48Z)
Host: 1.201.143.134:6280
Connection: Keep-Alive
Accept-Encoding: gzip

[.....A...-(.S.~8...#.
.!.{C{5!..}|..z=H.4..tM
&&...D0.91Qc.6#X..H....(.....5.}.g...cVI:.....e..Q".,n[.v.. &
6....^.....x..@2.6.s.5...T.....5r.9,..*~...d>.'Uum...WG.5... ..>u.\$..P<....
(....V...~.Z
a+.....U...4.u..w...Y...j...N..iI.....9.{/=..M..!
`..xw...=.P*.EW...&~BVMd.~.....>.^<..V. .#6.X.....~a.....!.'.|
TDK.....d..2.....T,.p....]....k....V7..+q...k..j.....:..?...7....9.h.
2..mr..=.JKf..7.....!!2S.#.....T.*.]... ..t.Z..

Packet 1. 1 client pkt(s), 0 server pkt(s), 0 turn(s). Click to select.

Entire conversation (779 bytes) Show and save data as ASCII Stream 0

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

IPS上的封包範例-2 (dst port 6280)

7

The screenshot shows a Wireshark interface with a single packet selected. The packet list pane shows:

No.	Time	Source	Report	Destination	Port	Protocol	Length	Info
1	0.000000	148.131.171.118	49568	123.56.205.151	6280	HTTP	843	POST: /collect?ts=1511948906000&ver=870403&diff=1456&hash=02017829714 HTTP/1.1 (application/octet-stream)

The 'Port' column value '6280' is circled in red. Below the packet list, the packet details pane shows the following structure:

- Frame 1: 843 bytes on wire (6744 bits), 843 bytes captured (6744 bits)
- Ethernet II, Src: Cisco_30:7d:32 (78:e4:22:30:7d:32), Dst: Cisco_31:d3:51 (78:e4:22:31:d3:51)
- 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 101
- Internet Protocol Version 4, Src: 148.131.171.118, Dst: 123.56.205.151
- Transmission Control Protocol, Src Port: 49568, Dst Port: 6280, Seq: 1, Ack: 1, Len: 785

The raw packet bytes are displayed at the bottom:

```
0000 70 e4 22 31 d3 51 70 e4 22 30 7d 32 81 00 00 65 p."1.Qp. "0}2...e
0010 88 00 45 00 03 39 76 5e 40 00 7b 06 05 97 8c 83 ..E..9v^ @_{.....
0020 ab 76 7b 38 cd 97 c1 a0 18 88 7f 37 cd 5e c1 a2 -v{B.... ..7.^..
0030 02 dd 50 18 00 ff 81 91 00 00 50 4f 53 54 20 2f ..P..... ..POST /
0040 63 6f 6c 6c 65 63 74 3f 74 73 3d 31 35 31 31 39 collect? ts=15119
0050 34 38 39 30 36 30 30 30 26 76 65 72 3d 38 37 30 48906000 &ver=870
0060 34 30 33 26 64 69 66 66 3d 31 34 35 36 26 68 61 4038diff =1456&ha
```

IPS上的封包範例-2 (符合特徵)

8

Wireshark · Follow TCP Stream (tcp.stream eq 0) · packet (1)

1. 2. 3. 4. 5.

```
POST /collect?ts=1511948906000&ver=870403&diff=1456&hash=02017829714 HTTP/1.1
Content-Length: 480
Content-Type: application/octet-stream
Charset: UTF-8
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.2; XT1052 Build/KOT49H)
Host: 123.56.205.151:6280
Connection: Keep-Alive
Accept-Encoding: gzip

d=...Z.2.....h...K.fK{.1..^(....(M.~...xq...]c,..Z.f.....6@~..?K....)....hc.AP..i
\,..@.....Iao..\V..I.B#b.?..nJ...]s...+.X..r.v.....#E...qz....T.SKZ=#...
6".;t....]B.~.Bm~.....QH.vK.;..R7..h...
...u      ...!R..Gs2c.y...5.2P...:z..y.....^..n.P@.....p.
.."{b.>...Ej...#.7E...8...../..)P..
.....[k..ek...r.%....}X."..dv.._[.....<.Mu.9d"...M...S\-.
...f.h.....K..3....W.w.AP..i\,..@.....Iao..\V..I.B#b.,...+L...b.H...<.xo.
$.e...5.*u?...L..z(.&..
```

1 client pkt(s), 0 server pkt(s), 0 turn(s).

Entire conversation (785 bytes) Show and save data as ASCII Stream 0

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

結論

9

- 此次Congur資安事件是符合IPS上的規則而開單；從IPS上調閱的封包，經過檢查後也都符合IPS的規則
- 似乎跟目標IP沒有太大的相關性(從該規則中只提到目標port)
- IPS無法檢查該APP權限的問題。

實測環境

10

- OS: Windows 10 Pro (1703版)
- Android模擬器: BlueStacks 3
- 測試遊戲: 無盡的邊疆 1.8.9 (非LINE版本)
- 測試時間: 約5分鐘
- 測試時的IP: 192.168.1.2 (筆電接網路線到AP, AP接出去的IP為140.112.3.250)

實測封包-1(無使用Filter)

11

The screenshot shows a Wireshark capture of network traffic. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, and Length. The packets include DNS queries and responses, TCP SYN and ACK packets, and HTTP GET requests. The bottom pane shows the details of the selected packet (No. 1), identifying it as an ARP request for the broadcast address.

No.	Time	Source	Destination	Protocol	Length	Info
18	1.510877	192.168.1.2	12687 172.217.27.147	80 TCP	54	12687 → 80 [ACK] Seq=711 Ack=382 Min=65792 Len=0
19	1.531377	192.168.1.2	56858 140.112.254.4	53 DNS	77	Standard query 0xb1f9 A eb.bluestacks.com
20	1.556453	192.168.1.2	56858 168.95.1.1	53 DNS	77	Standard query 0xb1f9 A eb.bluestacks.com
21	1.559209	168.95.1.1	53 192.168.1.2	56858 DNS	169	Standard query response 0xb1f9 A eb.bluestacks.com CNAME bluestacks-eb-prod.us-west-1.elasticbeanstalk.com A 52.52.15.181 A 54.183.3.119
22	1.562507	192.168.1.2	12688 52.52.15.181	80 TCP	66	12688 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
23	1.568842	140.112.254.4	53 192.168.1.2	56858 DNS	481	Standard query response 0xb1f9 A eb.bluestacks.com CNAME bluestacks-eb-prod.us-west-1.elasticbeanstalk.com A 52.52.15.181 A 54.183.3.119 NS ns-1.
24	1.592448	216.58.200.244	443 192.168.1.2	12204 TLSv1	475	Application Data
25	1.633421	192.168.1.2	12284 216.58.200.244	443 TCP	54	12284 → 443 [ACK] Seq=597 Ack=475 Min=258 Len=0
26	1.690487	52.52.15.181	80 192.168.1.2	12688 TCP	66	80 → 12688 [SYN, ACK] Seq=0 Ack=1 Min=26883 Len=0 MSS=1460 SACK_PERM=1 WS=256
27	1.690708	192.168.1.2	12688 52.52.15.181	80 TCP	54	12688 → 80 [ACK] Seq=1 Ack=1 Min=85536 Len=0
28	1.692173	192.168.1.2	12688 52.52.15.181	80 HTTP	185	GET /content_keymap/com.ekkorr.andlessfrontier.global.cfg/parseserver=2 HTTP/1.1
29	1.828222	52.52.15.181	80 192.168.1.2	12688 TCP	60	80 → 12688 [ACK] Seq=1 Ack=132 Win=28160 Len=0
30	1.824675	52.52.15.181	80 192.168.1.2	12688 HTTP	322	HTTP/1.1 404 NOT FOUND (text/html)
31	1.865411	192.168.1.2	12688 52.52.15.181	80 TCP	54	12688 → 80 [ACK] Seq=132 Ack=269 Min=65280 Len=0
32	1.999761	Broadcast	Broadcast	ARP	42	Who has 192.168.1.11? Tell 192.168.1.2
33	2.122650	192.168.1.2	12689 1.201.143.134	6280 TCP	66	12689 → 6280 [SYN] Seq=0 Min=65300 Len=0 MSS=1460 WS=256 SACK_PERM=1
34	2.202552	1.201.143.134	6280 192.168.1.2	12689 TCP	66	6280 → 12689 [SYN, ACK] Seq=0 Ack=1 Min=17920 Len=0 MSS=1460 SACK_PERM=1 WS=128
35	2.202649	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=1 Ack=1 Min=66560 Len=0
36	2.204273	192.168.1.2	12689 1.201.143.134	6280 HTTP	246	GET /plugin/170/version HTTP/1.1
37	2.284689	1.201.143.134	6280 192.168.1.2	12689 TCP	60	6280 → 12689 [ACK] Seq=1 Ack=193 Min=19072 Len=0
38	2.285653	1.201.143.134	6280 192.168.1.2	12689 TCP	285	6280 → 12689 [PSH, ACK] Seq=1 Ack=193 Min=19072 Len=231 [TCP segment of a reassembled PDU]
39	2.287273	1.201.143.134	6280 192.168.1.2	12689 HTTP	347	HTTP/1.1 200 OK (application/octet-stream)
40	2.287347	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=193 Ack=525 Min=66048 Len=0
41	2.293783	192.168.1.2	12689 1.201.143.134	6280 HTTP	268	GET /plugin/c/58/armvabi-v7a/RiskStub HTTP/1.1
42	2.303175	192.168.1.1	3072 239.255.255.250	1900 SSDP	307	NOTIFY * HTTP/1.1
43	2.304034	192.168.1.1	3072 239.255.255.250	1900 SSDP	379	NOTIFY * HTTP/1.1
44	2.304835	192.168.1.1	3072 239.255.255.250	1900 SSDP	375	NOTIFY * HTTP/1.1

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: AsustekC_96:99:2e (d8:50:a6:96:99:2e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

```
0000  ff ff ff ff ff ff d8 50 e6 96 99 2e 08 06 00 01  ....P.....
0010  08 00 06 04 00 01 d8 50 e6 96 99 2e c0 a8 01 02  ....P.....
0020  00 00 00 00 00 00 c0 a8 01 64  ....d
```

實測封包-2(無使用Filter)

12

The screenshot displays the Wireshark interface with a list of network packets. The top section shows a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The bottom section shows a detailed view of the selected packet (No. 42), which is an ARP request.

No.	Time	Source	Destination	Protocol	Length	Info
42	2.383175	192.168.1.1	3872 239.255.255.250	1900 SSDP	387	NOTIFY * HTTP/1.1
43	2.384834	192.168.1.1	3872 239.255.255.250	1900 SSDP	379	NOTIFY * HTTP/1.1
44	2.384835	192.168.1.1	3872 239.255.255.250	1900 SSDP	375	NOTIFY * HTTP/1.1
45	2.384835	192.168.1.1	3872 239.255.255.250	1900 SSDP	355	NOTIFY * HTTP/1.1
46	2.386836	192.168.1.1	3872 239.255.255.250	1900 SSDP	387	NOTIFY * HTTP/1.1
47	2.384836	192.168.1.1	3872 239.255.255.250	1900 SSDP	369	NOTIFY * HTTP/1.1
48	2.384837	192.168.1.1	3872 239.255.255.250	1900 SSDP	371	NOTIFY * HTTP/1.1
49	2.384837	192.168.1.1	3872 239.255.255.250	1900 SSDP	371	NOTIFY * HTTP/1.1
50	2.373435	1.281.143.134	6288 192.168.1.2	12689 TCP	290	6288 → 12689 [PSH, ACK] Seq=525 Ack=399 Win=20096 Len=236 [TCP segment of a reassembled PDU]
51	2.376428	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=761 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
52	2.374423	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=2221 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
53	2.374424	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=3681 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
54	2.374426	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=5141 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
55	2.374427	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=6601 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
56	2.376428	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=8061 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
57	2.374556	192.168.1.2	12689 1.281.143.134	6288 TCP	54	12689 → 6288 [ACK] Seq=399 Ack=9521 Win=66568 Len=0
58	2.376165	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=9521 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
59	2.376167	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=10981 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
60	2.376167	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=12441 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
61	2.376221	192.168.1.2	12689 1.281.143.134	6288 TCP	54	12689 → 6288 [ACK] Seq=399 Ack=13901 Win=66568 Len=0
62	2.454808	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=13901 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
63	2.455537	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [PSH, ACK] Seq=15361 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
64	2.455539	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=16821 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
65	2.455541	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=18281 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
66	2.455542	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=19741 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
67	2.455543	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=21201 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]
68	2.455543	1.281.143.134	6288 192.168.1.2	12689 TCP	1514	6288 → 12689 [ACK] Seq=22661 Ack=399 Win=20096 Len=1460 [TCP segment of a reassembled PDU]

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on Interface 0
Ethernet II, Src: AsustekC_96:99:2a (08:50:a6:96:99:2a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

```
0000 ff ff ff ff ff ff 08 50 a6 96 99 2a 08 00 00 00
0010 08 00 06 04 00 01 01 50 a6 96 99 2a c8 a8 91 02
0020 00 00 00 00 00 00 c8 a8 01 45
```

實測封包-3(設filter)

13

The screenshot displays the Wireshark interface with a list of captured network packets. The main pane shows a table of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 33) is highlighted in blue. Below the list, the packet details pane shows the structure of the selected packet: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. At the bottom, the packet bytes pane shows the raw hexadecimal and ASCII data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
716	4.659023	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=2159 Ack=806096 Win=725760 Len=0
721	7.458081	192.168.1.2	12689 1.201.143.134	6280 TCP	1514	12689 → 6280 [PSH, ACK] Seq=2159 Ack=806096 Win=725760 Len=1468 [TCP segment of a reassembled PDU]
722	7.512806	203.246.170.98	12689 192.168.1.2	6280 ICMP	78	Destination unreachable (Fragmentation needed)
724	7.538910	192.168.1.2	12689 1.201.143.134	6280 HTTP	61	POST /collect?ts=1511403991000&ver=636735&d1ff=2754&hash=01781240199 HTTP/1.1 (application/octet-stream)
728	7.663320	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=1626 Ack=808135 Win=725760 Len=0
729	8.189102	192.168.1.2	12689 1.201.143.134	6280 TCP	369	12689 → 6280 [PSH, ACK] Seq=3626 Ack=808135 Win=725760 Len=315 [TCP segment of a reassembled PDU]
730	8.189155	192.168.1.2	12689 1.201.143.134	6280 TCP	1514	12689 → 6280 [PSH, ACK] Seq=3941 Ack=808135 Win=725760 Len=1468 [TCP segment of a reassembled PDU]
732	8.269098	192.168.1.2	12689 1.201.143.134	6280 HTTP	258	POST /collect?ts=1511403991000&ver=636735&d1ff=1448&hash=01781240199 HTTP/1.1 (application/octet-stream)
737	8.410306	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=5605 Ack=808365 Win=725760 Len=0
1060	19.500039	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [FIN, ACK] Seq=5905 Ack=808365 Win=725760 Len=0
1072	19.867591	192.168.1.2	12689 1.201.143.134	6280 TCP	54	12689 → 6280 [ACK] Seq=5605 Ack=808366 Win=725760 Len=0
12587	69.866006	192.168.1.2	12748 1.201.143.134	6280 TCP	66	12748 → 6280 [SYN] Seq=0 Win=64620 Len=0 MSS=1460 WS=256 SACK_PERM=1
17109	69.716687	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=1 Ack=1 Win=66048 Len=0
17118	69.720812	192.168.1.2	12748 1.201.143.134	6280 HTTP	847	POST /collect?ts=1511403991000&ver=636735&d1ff=953&hash=01781240199 HTTP/1.1 (application/octet-stream)
17118	69.817710	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=794 Ack=231 Win=65792 Len=0
17417	130.818427	192.168.1.2	12748 1.201.143.134	6280 HTTP	832	POST /collect?ts=1511403991000&ver=636735&d1ff=1103&hash=01781240199 HTTP/1.1 (application/octet-stream)
17421	131.051337	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=1572 Ack=461 Win=65536 Len=0
18062	192.058607	192.168.1.2	12748 1.201.143.134	6280 HTTP	847	POST /collect?ts=1511403991000&ver=636735&d1ff=331&hash=01781240199 HTTP/1.1 (application/octet-stream)
18065	192.180023	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=2365 Ack=695 Win=65280 Len=0
18280	253.191358	192.168.1.2	12748 1.201.143.134	6280 HTTP	831	POST /collect?ts=1511403991000&ver=636735&d1ff=468&hash=01781240199 HTTP/1.1 (application/octet-stream)
18283	253.386109	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=3182 Ack=921 Win=65024 Len=0
18537	314.398658	192.168.1.2	12748 1.201.143.134	6280 HTTP	831	POST /collect?ts=1511403991000&ver=636735&d1ff=680&hash=01781240199 HTTP/1.1 (application/octet-stream)
18540	314.494083	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=3919 Ack=1151 Win=64768 Len=0
18829	375.498885	192.168.1.2	12748 1.201.143.134	6280 HTTP	847	POST /collect?ts=1511403991000&ver=636735&d1ff=775&hash=01781240199 HTTP/1.1 (application/octet-stream)
18837	375.594095	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=4712 Ack=1381 Win=64512 Len=0
19026	436.609758	192.168.1.2	12748 1.201.143.134	6280 HTTP	831	POST /collect?ts=1511403991000&ver=636735&d1ff=891&hash=01781240199 HTTP/1.1 (application/octet-stream)
19028	436.712668	192.168.1.2	12748 1.201.143.134	6280 TCP	54	12748 → 6280 [ACK] Seq=5489 Ack=1611 Win=66048 Len=0

Frame 33: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on Interface 0
Ethernet II, Src: AsusTek 96:99:2e (d8:5b:e6:96:99:2e), Dst: Zionsome-FB:a8:58 (78:44:76:fb:a8:58)
Internet Protocol Version 4, Src: 192.168.1.2, Dst: 1.201.143.134
Transmission Control Protocol, Src Port: 12689, Dst Port: 6280, Seq: 0, Len: 0

```
0000  78 44 76 fb a8 58 d8 50 e6 96 99 2e 08 00 45 00  .Dr..X.P.....E.  
0010  00 54 61 80 40 00 00 06 00 00 c0 a8 01 82 01 c9  .4a@.....  
0020  8f 86 31 91 18 88 48 74 97 74 00 00 00 00 80 02  .1...Ht.....  
0030  ff 3c 53 20 00 00 02 04 05 b4 01 03 03 08 01 01  <S.....  
0040  04 02  ..
```

觀察的結果

14

- 已知在遊玩遊戲時，平均1分鐘會丟出觸發資安事件的封包內容(如下圖，不論是沒有動作或是點擊遊戲內的icon)
- 目前有請廠商和原廠確認此規則的判斷細節，再另行回覆通知我們ASOC。

17110	69.720812	192.168.1.2	12748	1.201.143.134	6280	HTTP	847	POST	/collect?ts=1511403991000&ver=636735&diff=953&hash=01781240199	HTTP/1.1	(application/octet-stream)
17118	69.817710	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=794 Ack=231 Win=65792 Len=0		
17417	130.818427	192.168.1.2	12748	1.201.143.134	6280	HTTP	832	POST	/collect?ts=1511403991000&ver=636735&diff=1103&hash=01781240199	HTTP/1.1	(application/octet-stream)
17421	131.051337	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=3572 Ack=461 Win=65536 Len=0		
18062	192.058607	192.168.1.2	12748	1.201.143.134	6280	HTTP	847	POST	/collect?ts=1511403991000&ver=636735&diff=331&hash=01781240199	HTTP/1.1	(application/octet-stream)
18065	192.188023	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=2365 Ack=691 Win=65280 Len=0		
18280	253.191350	192.168.1.2	12748	1.201.143.134	6280	HTTP	831	POST	/collect?ts=1511403991000&ver=636735&diff=468&hash=01781240199	HTTP/1.1	(application/octet-stream)
18283	253.386109	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=3142 Ack=921 Win=65024 Len=0		
18537	314.398658	192.168.1.2	12748	1.201.143.134	6280	HTTP	831	POST	/collect?ts=1511403991000&ver=636735&diff=680&hash=01781240199	HTTP/1.1	(application/octet-stream)
18540	314.494083	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=3919 Ack=1151 Win=64768 Len=0		
18829	375.498885	192.168.1.2	12748	1.201.143.134	6280	HTTP	847	POST	/collect?ts=1511403991000&ver=636735&diff=775&hash=01781240199	HTTP/1.1	(application/octet-stream)
18837	375.594095	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=4712 Ack=1381 Win=64512 Len=0		
19026	436.609758	192.168.1.2	12748	1.201.143.134	6280	HTTP	831	POST	/collect?ts=1511403991000&ver=636735&diff=891&hash=01781240199	HTTP/1.1	(application/octet-stream)
19028	436.717468	192.168.1.2	12748	1.201.143.134	6280	TCP	54	12748	→ 6280 [ACK] Seq=5489 Ack=1611 Win=66048 Len=0		