Network Administration/System Administration Homework #4

Professor: Hsin-Mu Tsai

b03502040 劉君猷

Network Administration

1 DHCP

1.1 (10%)

```
dhclient -r interface && dhclient interface
```

REFERENCE:

(1) [http://www.cyberciti.biz/faq/howto-linux-renew-dhcp-client-ip-address/]

1.2 (15%)

It maybe that the client would request the previously used IP address from the DHCP server (except that the request is rejected by DHCP server for this IP is used by the other client or in the other sub-network), or DHCP server has its own table to remember past IP address assignments, it would get the same IP.

REFERENCE:

(1) [https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol]

(2) [http://unix.stackexchange.com/questions/131031/is-it-possible-for-dhcp-to-assign-different-ip-addresses-to-the-same-machine]

2 DNS

2.1 (15%)

1. Not handling-efficient compared with the distributed, for single DNS server doesn’t have different hierarchy of DNS server to handle different tier of sub-domain name parsing, it handles smaller number of requests in the same period.

2. Lower reliability. If a single DNS server is broken, all the DNS parsing is not working; whereas in the distributed situation, one DNS server is broken, only that DNS server is not working, the other servers can continue working their parts.

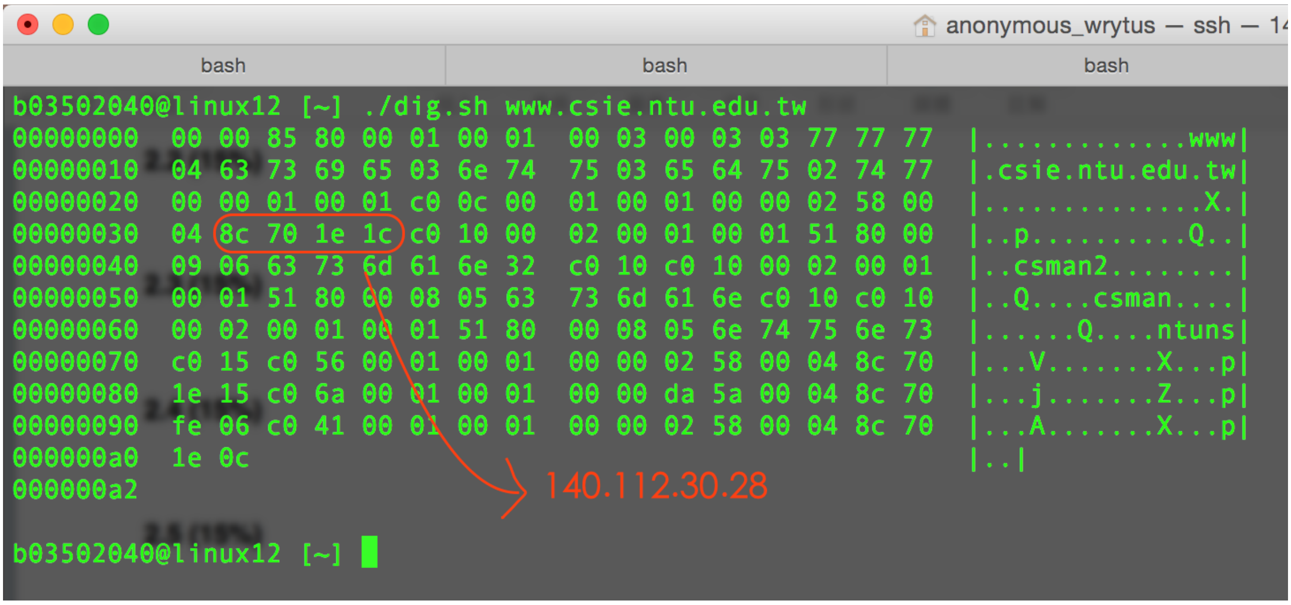
3. Bigger time delay. Distributed can be cached for the TLD DNS server so that it won't query the root DNS server every time. However, single DNS server cannot. The average query time is longer.

REFERENCE:

(1) [http://com2networks.blogspot.tw/2013/09/distributed-domain-name-system.html]

(2) [http://www-inf.int-evry.fr/~hennequi/CoursDNS/NOTES-COURS_eng/syst.html]

2.2 (15%)



```
b03502040@linux12 [~] ./dig.sh www.csie.ntu.edu.tw
00000000  00 00 85 80 00 01 00 01  00 03 00 03 03 77 77 77  |.............www|
00000010  04 63 73 69 65 03 6e 74  75 03 65 64 75 02 74 77  |.csie.ntu.edu.tw|
00000020  00 00 01 00 01 c0 0c 00  01 00 01 00 00 02 58 00  |..............X.|
00000030  04 8c 70 1e 1c c0 10 00  02 00 01 00 01 51 80 00  |..p..........Q..|
00000040  09 06 63 73 6d 61 6e 32  c0 10 c0 10 00 02 00 01  |..csman2........|
00000050  00 01 51 80 00 08 05 63  73 6d 61 6e c0 10 c0 10  |..Q....csman....|
00000060  00 02 00 01 00 01 51 80  00 08 05 6e 74 75 6e 73  |......Q....ntuns|
00000070  c0 15 c0 56 00 01 00 01  00 00 02 58 00 04 8c 70  |...V.......X...p|
00000080  1e 15 c0 6a 00 01 00 01  00 00 da 5a 00 04 8c 70  |...j.......Z...p|
00000090  fe 06 c0 41 00 01 00 01  00 00 02 58 00 04 8c 70  |...A.......X...p|
000000a0  1e 0c                                             |..|
000000a2

b03502040@linux12 [~]
```

140.112.30.28

2.3 (15%)

According to the hexdump of the response, we may construct a graph like the following graph. Whenever we see the 2-octet-length table begins with "1 1", we know that it's a pointer compressed by DNS. Then we can use the subsequent offset to trace back where the first appearance of the domain name is, so that achieving the decompression.

Because DNS use UDP by default. The features that UDP specified is small-sized and fast-transferring. If DNS doesn't compress the large response, the data size would too big to be transferred rapidly by UDP protocol.

REFERENCE:

(1) [http://publib.boulder.ibm.com/html/as400/v4r5/ic2979/info/RZAB6DNSFORMAT.HTM]

(2) [http://www.ccs.neu.edu/home/amislove/teaching/cs4700/fall09/handouts/project1-primer.pdf]

```
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 20 |           1           |           F           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 22 |           3           |           I           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 24 |           S           |           I           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 26 |           4           |           A           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 28 |           R           |           P           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 30 |           A           |           0           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
...
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 40 |           3           |           F           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 42 |           0           |           0           |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 44 | 1  1|                20                       |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
...
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 64 | 1  1|                26                       |
    +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

2.4 (15%)

Distributed DoS attack. If we don't limit the query from NTU network, attacker can spoof the source IP address maliciously and make this query to DNS. Then the response from DNS server will be amplified and sent to the victim IP address, which cause a heavy traffic on bandwidth.

REFERENCE:

(1) [http://www.networkworld.com/article/2886283/security0/top-10-dns-attacks-likely-to-infiltrate-your-network.html#slide1]

2.5 (15%)

Which is dig_new.sh

REFERENCE:

(1) [http://www.tldp.org/LDP/abs/html/internalvariables.html]

2.6 (Bonus 10%)

Which is dig_bonus.sh

REFERENCE:

(1) [http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4]