

NASA 2022 Homework1

Network Administration

看個影集也會不小心洩漏密碼

找不到含有帳號密碼的封包，因為HTTPS雖是透過HTTP進行通訊，但會先使用SSL/TLS加密之後才會傳輸。

農場危機

1. 我利用Wireshark將封包過濾，選擇只看HTTP，並將內容下載之後發現有一個QR code，其他都是佩佩豬。
2. 我另外有發現pig這個html檔，並且在封包內找到<http://nasahw1.csie.ntu.edu.tw/pig/encode.js>，我將其複製下來，改寫了一下pig的內容，使QR code可以透過呼叫encode.js內的convertImage()這個function去修復；得到完整得QR code之後，掃描即得密碼為 NASA{2022_pig_pig
}。

這麼多網路協定

1. ICMP request:

```
→ 8376 44.723927 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=0/0, ttl=64 (rep)
← 8377 44.728413 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=0/0, ttl=57 (req)
8383 45.724311 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=1/256, ttl=64 (req)
8384 45.736043 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=1/256, ttl=57 (req)
8387 46.724940 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=2/512, ttl=64 (req)
8388 46.735230 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=2/512, ttl=57 (req)
8395 47.726161 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=3/768, ttl=64 (req)
8396 47.729914 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=3/768, ttl=57 (req)
8445 48.729707 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=4/1024, ttl=64 (req)
8446 48.734642 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=4/1024, ttl=57 (req)
8463 49.734352 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=5/1280, ttl=64 (req)
8464 49.744118 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=5/1280, ttl=57 (req)

> Internet Protocol Version 4, Src: 192.168.31.103, Dst: 172.217.160.68
`- Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xe24e [correct]
    [Checksum Status: Good]
    Identifier (BE): 16412 (0x401c)
    Identifier (LE): 7232 (0x1c40)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
    [Response frame: 8377]
    Timestamp from icmp data: Feb 28, 2022 20:34:03.771886000 CST
    [Timestamp from icmp data (relative): 0.000041000 seconds]
  ` Data (48 bytes)
    0000 64 09 80 7b b4 31 14 7d da 0e c9 d9 08 00 45 00 d...{.1} .....E.
    0010 00 54 29 37 00 00 40 01 24 45 c0 a8 1f 67 ac d9 T)7 @. $E...g...
    0020 a0 44 08 00 e2 4e 40 1c 00 00 62 1c c1 3b 00 0b .D..N@. ..b..;..
    0030 c7 2e 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 . .....
    0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 . .....
    0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*,,- ./012345
    0060 36 37                                         67


```

ICMP reply:

```
→ 8376 44.723927 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=0/0, ttl=64 (rep)
← 8377 44.728413 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=0/0, ttl=57 (req)
8383 45.724311 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=1/256, ttl=64 (req)
8384 45.736043 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=1/256, ttl=57 (req)
8387 46.724940 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=2/512, ttl=64 (req)
8388 46.735230 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=2/512, ttl=57 (req)
8395 47.726161 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=3/768, ttl=64 (req)
8396 47.729914 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=3/768, ttl=57 (req)
8445 48.729707 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=4/1024, ttl=64 (req)
8446 48.734642 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=4/1024, ttl=57 (req)
8463 49.734352 192.168.31.103 172.217.160.68 ICMP 98 Echo (ping) request id=0x401c, seq=5/1280, ttl=64 (req)
8464 49.744118 172.217.160.68 192.168.31.103 ICMP 98 Echo (ping) reply id=0x401c, seq=5/1280, ttl=57 (req)

`- Internet Control Message Protocol
  Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xea4e [correct]
    [Checksum Status: Good]
    Identifier (BE): 16412 (0x401c)
    Identifier (LE): 7232 (0x1c40)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
    [Request frame: 8376]
    [Response time: 4.486 ms]
    Timestamp from icmp data: Feb 28, 2022 20:34:03.771886000 CST
    [Timestamp from icmp data (relative): 0.004527000 seconds]
  ` Data (48 bytes)
    0000 14 7d da 0e c9 d9 64 09 80 7b b4 31 08 00 45 00 .}....d...{.1..E.
    0010 00 54 00 00 00 00 39 01 54 7c ac d9 a0 44 c0 a8 T...9. T|..D..
    0020 1f 67 00 00 ea 4e 40 1c 00 00 62 1c c1 3b 00 0b g..N@. ..b..;..
    0030 c7 2e 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 . .....
    0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 . .....
    0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*,,- ./012345
    0060 36 37                                         67


```

ICMP最主要的是用來解析網路封包或是分析路由，大多透過回傳回來的錯誤訊息進行分析，屬於Network layer。

2. DNS query:

76	16.387685	192.168.31.103	192.168.31.1	DNS	72	Standard query 0x4c35 A TPEA90103943
77	16.397666	192.168.31.1	192.168.31.103	DNS	72	Standard query response 0x4c35 A TPEA90103943
146	24.625529	192.168.31.103	192.168.31.1	DNS	79	Standard query 0xdd40 HTTPS clients1.google.com
147	24.625629	192.168.31.103	192.168.31.1	DNS	79	Standard query 0xc833 A clients1.google.com
148	24.657496	192.168.31.1	192.168.31.103	DNS	129	Standard query response 0xc833 A clients1.google.com
149	24.657504	192.168.31.1	192.168.31.103	DNS	153	Standard query response 0xdd40 HTTPS clients1.google.com
150	24.658240	192.168.31.103	192.168.31.1	DNS	80	Standard query 0xb8da HTTPS clients.l.google.com
155	24.699779	192.168.31.1	192.168.31.103	DNS	130	Standard query response 0xb8da HTTPS clients.l.google.com
334	28.305398	192.168.31.103	192.168.31.1	DNS	90	Standard query 0x801d HTTPS api-glb-apne1c.smoothapp.net
335	28.305467	192.168.31.103	192.168.31.1	DNS	90	Standard query 0x64fa A api-glb-apne1c.smoothapp.net
337	28.309630	192.168.31.1	192.168.31.103	DNS	155	Standard query response 0x64fa A api-glb-apne1c.smoothapp.net
338	28.310022	192.168.31.103	192.168.31.1	DNS	95	Standard query 0x2fa0 HTTPS smooth-searchv2-apne1c.smoothapp.net
339	28.312077	192.168.31.1	192.168.31.103	DNS	191	Standard query response 0x801d HTTPS api-glb-apne1c.smoothapp.net

```

> Frame 76: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface en0, id 0
> Ethernet II, Src: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9), Dst: XiaomiCo_7b:b4:31 (64:09:80:7b:b4:31)
> Internet Protocol Version 4, Src: 192.168.31.103, Dst: 192.168.31.1
> User Datagram Protocol, Src Port: 61216, Dst Port: 53
> Domain Name System (query)

0000  64 09 80 7b b4 31 14 7d da 0e c9 d9 08 00 45 00  d...{.1}....E.
0010  00 3a 02 96 00 00 40 11 b8 64 c0 a8 1f 67 c0 a8  :...@.d..g...
0020  1f 01 ef 20 00 35 00 26 98 ef 4c 35 01 00 00 01  ....5.&.L5...
0030  00 00 00 00 00 00 0c 54 50 45 41 39 30 31 30 33  .....T PEA90103
0040  39 34 33 00 00 01 00 01                           943.....

```

DNS response:

76	16.387685	192.168.31.103	192.168.31.1	DNS	72	Standard query 0x4c35 A TPEA90103943
77	16.397666	192.168.31.1	192.168.31.103	DNS	72	Standard query response 0x4c35 A TPEA90103943
146	24.625529	192.168.31.103	192.168.31.1	DNS	79	Standard query 0xdd40 HTTPS clients1.google.com
147	24.625629	192.168.31.103	192.168.31.1	DNS	79	Standard query 0xc833 A clients1.google.com
148	24.657496	192.168.31.1	192.168.31.103	DNS	129	Standard query response 0xc833 A clients1.google.com
149	24.657504	192.168.31.1	192.168.31.103	DNS	153	Standard query response 0xdd40 HTTPS clients1.google.com
150	24.658240	192.168.31.103	192.168.31.1	DNS	80	Standard query 0xb8da HTTPS clients.l.google.com
155	24.699779	192.168.31.1	192.168.31.103	DNS	130	Standard query response 0xb8da HTTPS clients.l.google.com
334	28.305398	192.168.31.103	192.168.31.1	DNS	90	Standard query 0x801d HTTPS api-glb-apne1c.smoothapp.net
335	28.305467	192.168.31.103	192.168.31.1	DNS	90	Standard query 0x64fa A api-glb-apne1c.smoothapp.net
337	28.309630	192.168.31.1	192.168.31.103	DNS	155	Standard query response 0x64fa A api-glb-apne1c.smoothapp.net
338	28.310022	192.168.31.103	192.168.31.1	DNS	95	Standard query 0x2fa0 HTTPS smooth-searchv2-apne1c.smoothapp.net
339	28.312077	192.168.31.1	192.168.31.103	DNS	191	Standard query response 0x801d HTTPS api-glb-apne1c.smoothapp.net

```

> Frame 77: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface en0, id 0
> Ethernet II, Src: XiaomiCo_7b:b4:31 (64:09:80:7b:b4:31), Dst: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
> Internet Protocol Version 4, Src: 192.168.31.1, Dst: 192.168.31.103
> User Datagram Protocol, Src Port: 53, Dst Port: 61216
> Domain Name System (response)

0000  14 7d da 0e c9 d9 64 09 80 7b b4 31 08 00 45 00  }...d...{.1}....E.
0010  00 3a 00 00 40 00 40 11 7a fa c0 a8 1f 01 c0 a8  :...@.z.....
0020  1f 67 00 35 ef 20 00 26 18 6f 4c 35 81 80 00 01  g.5.&.oL5...
0030  00 00 00 00 00 00 0c 54 50 45 41 39 30 31 30 33  .....T PEA90103
0040  39 34 33 00 00 01 00 01                           943.....

```

DNS分為client與server，client會向server發query去問一個domain name的真正IP位址，而server必須response，詢問方法可分recursive與iterative。DNS是屬application layer。

3. ARP request:

2 0.000627	Apple_0e:c9:d9	ae:49:db:14:d6:64	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
5 0.005711	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
6 0.006057	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
14 0.378787	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
17 0.701108	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
18 0.702892	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
83 4.064331	Apple_0e:c9:d9	ae:49:db:14:d6:64	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
86 4.074296	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
87 4.075180	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
98 4.284485	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
99 4.286897	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
108 4.400838	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
152 4.724743	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5

> Frame 2: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
> Ethernet II, Src: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9), Dst: ae:49:db:14:d6:64 (ae:49:db:14:d6:64)
 ▾ Address Resolution Protocol (request)
 Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: request (1)
 Sender MAC address: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
 Sender IP address: 172.20.10.5
 Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
 Target IP address: 172.20.10.1

0000	ae	49	db	14	d6	64	14	7d	da	0e	c9	d9	08	06	00	01	·I···d···
0010	08	00	06	04	00	01	14	7d	da	0e	c9	d9	ac	14	0a	05
0020	00	00	00	00	00	00	ac	14	0a	01

ARP reply:

2 0.000627	Apple_0e:c9:d9	ae:49:db:14:d6:64	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
5 0.005711	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
6 0.006057	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
14 0.378787	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
17 0.701108	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
18 0.702892	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
83 4.064331	Apple_0e:c9:d9	ae:49:db:14:d6:64	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
86 4.074296	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
87 4.075180	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
98 4.284485	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5
99 4.286897	ae:49:db:14:d6:64	Apple_0e:c9:d9	ARP	42 172.20.10.1 is at ae:49:db:14:d6:64
108 4.400838	Apple_0e:c9:d9	Broadcast	ARP	42 ARP Announcement for 172.20.10.5
152 4.724743	Apple_0e:c9:d9	Broadcast	ARP	42 Who has 172.20.10.1? Tell 172.20.10.5

> Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
> Ethernet II, Src: ae:49:db:14:d6:64 (ae:49:db:14:d6:64), Dst: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
 ▾ Address Resolution Protocol (reply)
 Hardware type: Ethernet (1)
 Protocol type: IPv4 (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: reply (2)
 Sender MAC address: ae:49:db:14:d6:64 (ae:49:db:14:d6:64)
 Sender IP address: 172.20.10.1
 Target MAC address: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
 Target IP address: 172.20.10.5

0000	14	7d	da	0e	c9	d9	ae	49	db	14	d6	64	08	06	00	01	·}....I...d...
0010	08	00	06	04	00	02	ae	49	db	14	d6	64	ac	14	0a	05I...d...
0020	14	7d	da	0e	c9	d9	ac	14	0a	01

ARP是網址分析協定，以查詢的方式來獲得IP位址和實體位址的對應關係。其屬於Network layer與Link layer。當主機A有一個封包要給主機B，並且已獲得主機B的IP位址，那主機A會先檢查自己的ARP表格中有沒有該IP位址相對應的實體位址。如果有的話就將此IP所對應之MAC address填入layer 2的表頭；沒有的話就向網路發出ARP request的廣播封包，查詢主機B的實體位址。

4. DHCP discover:

2	0.000564	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xd99a459f
4	0.050419	172.20.10.1	172.20.10.5	DHCP	342	DHCP Offer - Transaction ID 0xd99a459f
5	0.050748	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xd99a459f
6	0.054180	172.20.10.1	172.20.10.5	DHCP	342	DHCP ACK - Transaction ID 0xd99a459f

> Frame 2: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 68, Dst Port: 67
 > Dynamic Host Configuration Protocol (Discover)
 Message type: Boot Request (1)
 Hardware type: Ethernet (0x01)
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0xd99a459f
 Seconds elapsed: 0
 > Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0
 Your (client) IP address: 0.0.0.0

DHCP offer:

4	0.050419	172.20.10.1	172.20.10.5	DHCP	342	DHCP Offer - Transaction ID 0xd99a459f
5	0.050748	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xd99a459f
6	0.054180	172.20.10.1	172.20.10.5	DHCP	342	DHCP ACK - Transaction ID 0xd99a459f

> Frame 4: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en0, id 0
 > Ethernet II, Src: ae:49:db:14:d6:64 (ae:49:db:14:d6:64), Dst: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
 > Internet Protocol Version 4, Src: 172.20.10.1, Dst: 172.20.10.5
 > User Datagram Protocol, Src Port: 67, Dst Port: 68
 > Dynamic Host Configuration Protocol (Offer)
 Message type: Boot Reply (2)
 Hardware type: Ethernet (0x01)
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0xd99a459f
 Seconds elapsed: 0
 > Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0
 Your (client) IP address: 172.20.10.5

DHCP request:

5	0.050748	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xd99a459f
6	0.054180	172.20.10.1	172.20.10.5	DHCP	342	DHCP ACK - Transaction ID 0xd99a459f

> Frame 5: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
 > User Datagram Protocol, Src Port: 68, Dst Port: 67
 > Dynamic Host Configuration Protocol (Request)
 Message type: Boot Request (1)
 Hardware type: Ethernet (0x01)
 Hardware address length: 6
 Hops: 0
 Transaction ID: 0xd99a459f
 Seconds elapsed: 1
 > Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0
 Your (client) IP address: 0.0.0.0

DHCP ACK:

L 6 0.054180 172.20.10.1 172.20.10.5 DHCP 342 DHCP ACK - Transaction ID 0xd99a459f

> Frame 6: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface en0, id 0
> Ethernet II, Src: ae:49:db:14:d6:64 (ae:49:db:14:d6:64), Dst: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
> Internet Protocol Version 4, Src: 172.20.10.1, Dst: 172.20.10.5
> User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (ACK)
Message type: Boot Reply (2)
Hardware type: Ethernet (0x01)
Hardware address length: 6
Hops: 0
Transaction ID: 0xd99a459f
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0
Your (client) IP address: 172.20.10.5
Next server IP address: 172.20.10.1
Relay agent IP address: 0.0.0.0
Client MAC address: Apple_0e:c9:d9 (14:7d:da:0e:c9:d9)
Client hardware address padding: 00000000000000000000000000000000

DHCP協定主要是為了節省子網路中IP位址的使用量，可以設定網路中的一台主機做為DHCP server，負責動態的分配IP位址，當網路中有任何電腦需要連線，才會向DHCP server要求一個IP位址，此時DHCP server會在資料庫中找一個未被使用的IP位址給該電腦使用，直到使用結束之後再將此IP位址歸還給DHCP server。

DHCP是位於Application layer。

System Administration

Permission

I. Basic

1. **False**，要 `ls` 一個資料夾，必須對其有 `r` 的權限。
 2. **True**，因為讀一個檔案的基本權限是 `user` 對該檔案所在目錄具有 `x` 權限，且對該檔案本身有 `r` 權限。
 3. **True**，原因同上題，因為對 `./dir1` 具有 `x` 權限且對 `./dir/dir3` 有 `r` 權限所以可以。
 4. **True**，只要對某個目錄具有 `x` 權限即可進入該目錄。
 5. **False**，因為對 `dir2` 沒有 `x` 權限。
 6. **False**，修改一個檔案至少需要對該檔案所在目錄擁有 `x` 且對該檔案本身具有 `w` 的權限。
 7. **False**，因為對 `fileB` 沒有 `w` 的權限。
 8. **True**，如上敘述，對 `dir1` 有 `x` 且對 `fileC` 有 `w` 的權限，因此可以。
 9. **True**，在某目錄新增或刪除檔案，等於是修改該目錄，因此要對該目錄具有 `w` 且對其所在目錄具有 `x` 權限。

10. **False**，對 dir4 沒有 w 所以沒辦法刪除其內部的檔案。

II. ACL

1. 透過 `getfacl` 可以看到 flags 為 `--t` 代表 sticky bit 有 on，所以只有資料夾擁有者或 super user 或是擁有他們的權限的人可以刪除裡面的檔案。
2. 透過 `setfacl -m g:ta:rx b08902149` 以及 `chmod 0700 b08902149` 來將該資料夾改成只有自己與 group ta 可檢視，另外也要 `setfacl -m m:rwx b08902149` 防止 mask 將某些權限擋掉。
3.
 - a. 我使用 b09505014 當好朋友
 - b. 進到 b08902149，先用 `setfacl -m u:b09505014:x` 將 b08902149 這個資料夾改成讓好友可以進入。接著以 `setfacl -m u:b09505014:rwx chatroom`、`setfacl -m g::--- chatroom/` 以及 `setfacl -m o::--- chatroom/` 將 chatroom 設成只有我和好朋友可以新增、刪除檔案以及檢視。另外也要 `setfacl -m m:rwx chatroom` 才能防止 mask 將某些權限擋掉。
 - c. 接續上一題，用 `setfacl -m d:u:b09505014:rwx chatroom` 將好友的設定繼承到 chatroom 以下的所有檔案，這樣就可以使之後新增的檔案都可以被雙方編輯。
 - d. 透過 `setfacl -m u::rx chatroom` 將自己設成只可檢視，並透過 `setfacl -m m:rx chatroom` 來讓好朋友只留下 rx 權限。
4.
 - a. 我透過 `setfacl -m g:ta:x wordle` 來讓 ta 可以進入 wordle，然後用 `chmod 0700 game.sh` 以及 `setfacl -m g:ta:rw game.sh` 使得只有 ta 對 game.sh 有讀寫權限（`chmod` 要在 `setfacl` 之前使用，如果在 `setfacl` 後才用 `chmod` 會變成在改 ACL 的 mask）。
 - b. 透過 `chmod 0400 wordlist.txt` 即可。
 - c. 如果一個執行檔 setuid 有 on，就是讓其他使用者在執行這個執行檔時，effective uid 會轉成該執行檔的 owner 之權限，此時就可以用該 owner 的權限去執行指令。如果要讓 shell script 有 setuid 的效果，就要去對 `bin/sh` setuid（假設該 shell script 第一行是 `#!/bin/sh`），那這樣可能會造成 race condition，例如有其他使用者在你所執行的 shell script 還沒結束前也去使用 `/bin/sh` 來執行一些操作（且此時也剛好 context switch 到他），此時他就可以用你的權限去做一些很危險的操作，例如去修改一些只有你有權限的檔案。