

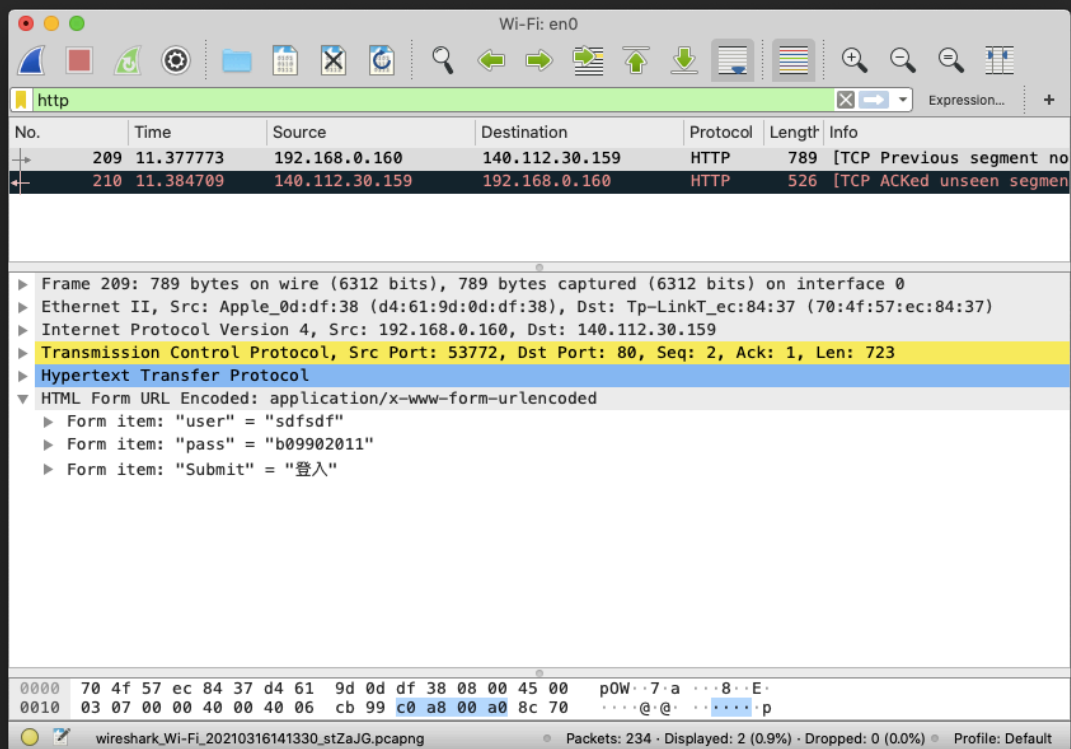
NASA HW1

Author: B09902011 陳可邦

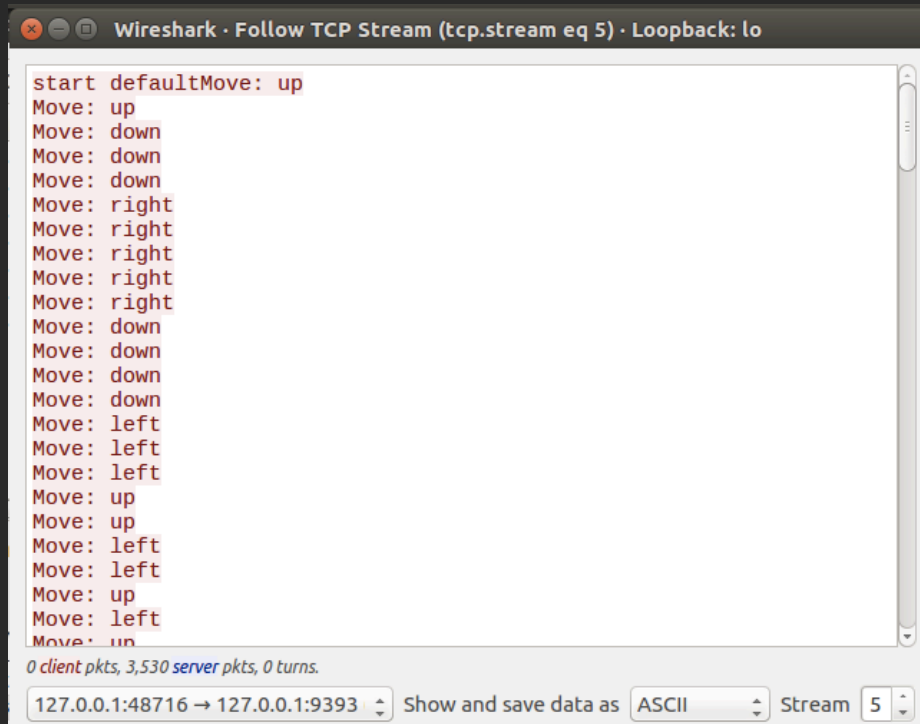
NA

野生的密碼難道會在網路上赤裸地奔馳著？

1. We can simply filter by http and get this, since http don't do encryption:

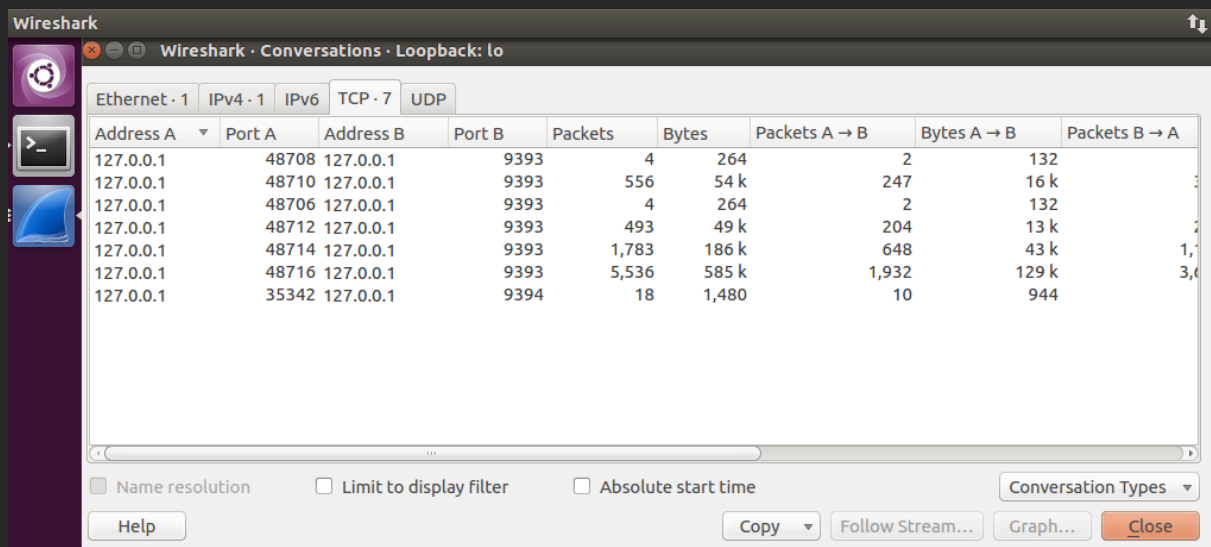


2. This version uses https, which encrypts requests & responses, so we can't get our password from Wireshark.

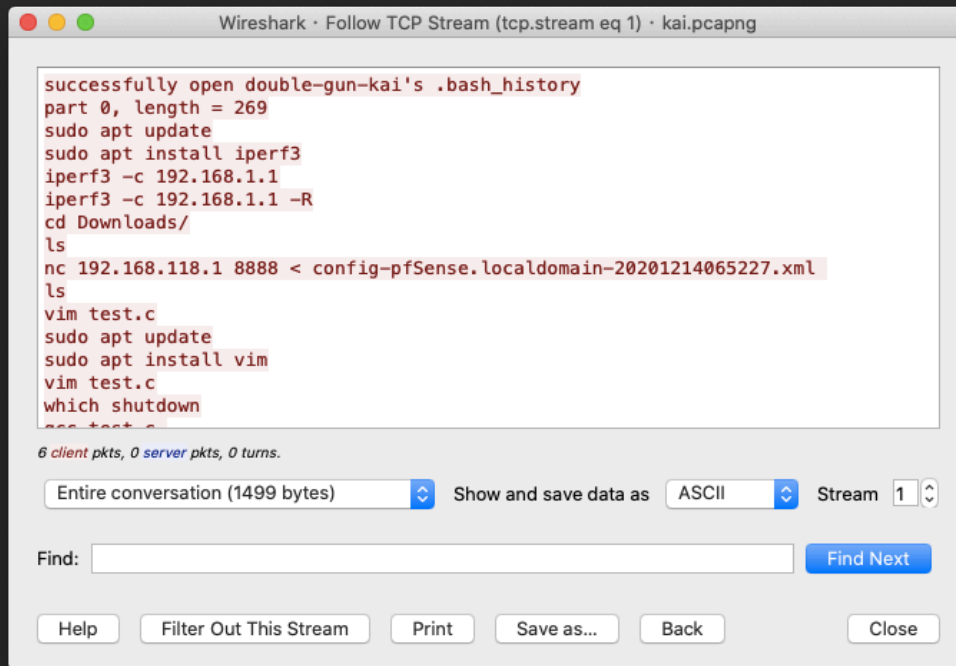


So basically, the game server sent us the time, the ball and the pad's location, and the client send directions when we press move.

2. If we pay attention to the conversations page, we can notice at some point there's another conversation happening on port 9394:

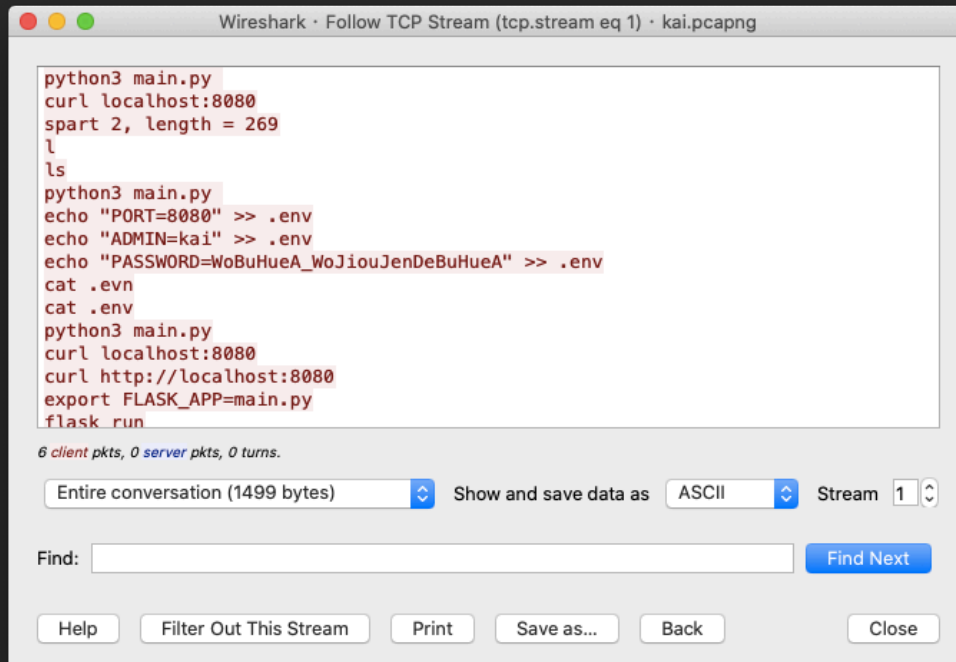


And when we look into it we'll discover this:



So we know the game steals our .bash_history.

3. We repeat the above steps with the pcap:



PASSWORD=WoBuHueA_WoJiouJenDeBuHueA

4. I wrote a quick python program to communicate with the server, since we already know the port and the format:

```

import socket
host = '127.0.0.1'
port = 9393

with socket.socket(socket.AF_INET,socket.SOCK_STREAM) as s:
    s.connect((host,port))
    print('connected')
    s.sendall(b'start fast')
    while True:
        data = s.recv(1024).decode('ascii')
        if 'hori' not in data:
            # Stop when recieving anything not game data
            print(data)
            break
        else:
            data=data.split('\n')
            # Make sure it don't explode when server asks for secret
            plus = 0
            if 'secret' in data[0]:
                plus = 1

            x=int(data[0+plus].split(' ')[-1])
            y=int(data[1+plus].split(' ')[-1])
            bx=int(data[2+plus].split(' ')[-1])
            by=int(data[3+plus].split(' ')[-1])
            t=int(data[4+plus].split(' ')[-1])

            if x<bx: s.sendall(b'Move: right');print('r')
            elif x>bx: s.sendall(b'Move: left');print('l')
            elif y>by: s.sendall(b'Move: up');print('u')
            elif y<by: s.sendall(b'Move: down');print('d')

```

Got the flag: `HW1{d0_y0u_knovv_wH0_KaienLin_1s?}`

The image shows a Wireshark packet capture of ICMP Echo (ping) messages. The top pane displays a list of 8 packets (No. 31-42) with columns for No., Time, Source, Destination, Protocol, Length, and Info. The bottom pane shows the detailed view of packet 32, which is an Echo (ping) reply from 172.217.160.78 to 192.168.0.160. The packet details include Ethernet II, Internet Protocol Version 4, and Internet Control Message Protocol (ICMP) Type: 0 (Echo (ping) reply). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
31	6.612337	192.168.0.160	172.217.160.78	ICMP	98	Echo (ping) request id=0xe9f7, seq=0/0, ttl=64 (reply in 32)
32	6.616232	172.217.160.78	192.168.0.160	ICMP	98	Echo (ping) reply id=0xe9f7, seq=0/0, ttl=115 (request in 31)
33	7.613444	192.168.0.160	172.217.160.78	ICMP	98	Echo (ping) request id=0xe9f7, seq=1/256, ttl=64 (reply in 34)
34	7.620055	172.217.160.78	192.168.0.160	ICMP	98	Echo (ping) reply id=0xe9f7, seq=1/256, ttl=115 (request in 33)
35	8.618734	192.168.0.160	172.217.160.78	ICMP	98	Echo (ping) request id=0xe9f7, seq=2/512, ttl=64 (reply in 36)
36	8.624102	172.217.160.78	192.168.0.160	ICMP	98	Echo (ping) reply id=0xe9f7, seq=2/512, ttl=115 (request in 35)
41	9.624015	192.168.0.160	172.217.160.78	ICMP	98	Echo (ping) request id=0xe9f7, seq=3/768, ttl=64 (reply in 42)
42	9.632007	172.217.160.78	192.168.0.160	ICMP	98	Echo (ping) reply id=0xe9f7, seq=3/768, ttl=115 (request in 41)

Frame 32: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: Tp-LinkT_ec:84:37 (70:4f:57:ec:84:37), Dst: Apple_0d:df:38 (d4:61:9d:0d:df:38)
 Internet Protocol Version 4, Src: 172.217.160.78, Dst: 192.168.0.160
 Internet Control Message Protocol
 Type: 0 (Echo (ping) reply)
 Code: 0
 Checksum: 0x2ac6 [correct]
 [Checksum Status: Good]
 Identifier (BE): 59895 (0xe9f7)

0020 00 a0 00 00 2a c6 e9 f7 00 00 60 50 6a 7c 00 03*...Pj...
 0030 35 6f 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 5o.....
 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 6'()*+,-./012345
 0060 36 3767

1. ICMP, or the Internet Control Message Protocol is used to send error messages and other informations, for figuring out networking issues. It is built on the Network Layer.

The image shows a Wireshark packet capture of DNS queries and responses. The top pane displays a list of 12 packets (No. 50-101) with columns for No., Time, Source, Destination, Protocol, Length, and Info. The bottom pane shows the detailed view of packet 101, which is a Standard query response from 192.168.0.160 to 192.168.0.1. The packet details include Ethernet II, Internet Protocol Version 4, and DNS. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
50	11.331864	192.168.0.160	192.168.0.1	DNS	76	Standard query 0x993a A zh.wikipedia.org
51	11.338792	192.168.0.1	192.168.0.160	DNS	121	Standard query response 0x993a A zh.wikipedia.org CNAME dyna.wikim
111	13.774168	192.168.0.160	192.168.0.1	DNS	80	Standard query 0x9472 A upload.wikimedia.org
112	13.778874	192.168.0.1	192.168.0.160	DNS	96	Standard query response 0x9472 A upload.wikimedia.org A 103.102.16
972	14.777480	192.168.0.160	192.168.0.1	DNS	81	Standard query 0x2197 A commons.wikimedia.org
973	14.781081	192.168.0.160	192.168.0.1	DNS	78	Standard query 0xbe93 A meta.wikimedia.org
974	14.783281	192.168.0.1	192.168.0.160	DNS	116	Standard query response 0x2197 A commons.wikimedia.org CNAME dyna.
976	14.787434	192.168.0.1	192.168.0.160	DNS	113	Standard query response 0xbe93 A meta.wikimedia.org CNAME dyna.wik
977	14.790569	192.168.0.160	192.168.0.1	DNS	76	Standard query 0x0203 A wikiplus-app.com
997	14.934887	192.168.0.1	192.168.0.160	DNS	108	Standard query response 0x0203 A wikiplus-app.com A 167.88.177.22
1011	15.081406	192.168.0.160	192.168.0.1	DNS	76	Standard query 0x956a A en.wikipedia.org
1014	15.087085	192.168.0.1	192.168.0.160	DNS	121	Standard query response 0x956a A en.wikipedia.org CNAME dyna.wikim

Answer RRs: 2
 Authority RRs: 0
 Additional RRs: 0
 Queries
 Answers
 en.wikipedia.org: type CNAME, class IN, cname dyna.wikimedia.org
 dyna.wikimedia.org: type A, class IN, addr 103.102.166.224
 [Request In: 1011]
 [Time: 0.005679000 seconds]

0000 d4 61 9d 0d df 38 70 4f 57 ec 84 37 08 00 45 00 .a...8p0 W...7..E.
 0010 00 6b 00 00 40 00 3f 11 b9 90 c0 a8 00 01 c0 a8 .k...@.7.....
 0020 00 a0 00 35 ed 9a 00 57 07 eb 95 6a 81 80 00 01 ...5...W...j....
 0030 00 02 00 00 00 00 02 65 6e 09 77 69 6b 69 70 65e n-wikipi
 0040 64 69 61 03 6f 72 67 00 00 01 00 01 c0 0c 00 05 dia.org.....

2. DNS protocol is used to communicate with DNS servers, and figure out a domain's IP address. It is built on the Application Layer.

4. DHCP protocol, or Dynamic Host Configuration Protocol, is used to communicate with DHCP servers when connecting to it. It handles the process of getting an IP address inside a network.