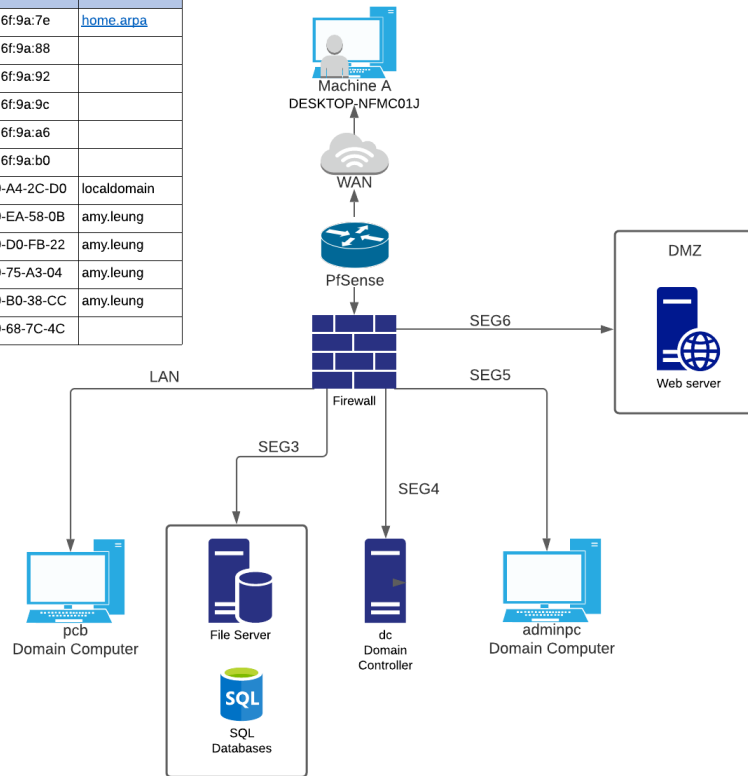


## Network Topology

Device Name	Interface	IP Address	MAC Address	Network Name
pfsense	WAN	192.168.1.128 /24	00:0c:29:6f:9a:7e	<a href="#">home.arpa</a>
	LAN	192.168.2.1/24	00:0c:29:6f:9a:88	
	SEG3	192.168.3.1/24	00:0c:29:6f:9a:92	
	SEG4	192.168.4.1/24	00:0c:29:6f:9a:9c	
	SEG5	192.168.5.1/24	00:0c:29:6f:9a:a6	
	SEG6	192.168.6.1/24	00:0c:29:6f:9a:b0	
DESKTOP-NFMC01J	WAN	192.168.1.129 /24	00-0C-29-A4-2C-D0	localdomain
pcb	LAN	192.168.2.30/24	00-0C-29-EA-58-0B	amy.leung
fileservr	SEG3	192.168.3.25/24	00-0C-29-D0-FB-22	amy.leung
dc	SEG4	192.168.4.30/24	00-0C-29-75-A3-04	amy.leung
adminpc	SEG5	192.168.5.30/24	00-0C-29-B0-38-CC	amy.leung
webserver	SEG6	192.168.6.25/24	00-0C-29-68-7C-4C	



This is the IT Security Final project

#### A. Accessing internal website

As background, I am accessing the webserver site (192.168.6.25) using the fileserver (192.168.3.25).

#### SC#1 – screenshot of accessing the webserver site

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.3.25	192.168.5.30	NBSS	55	NBSS Continuation Message
2	0.000517	192.168.5.30	192.168.3.25	TCP	66	49756 → 445 [ACK] Seq=1 Ack=2 Win=8210 Len=0 SLE=1 S
3	0.192828	192.168.5.30	192.168.3.25	TCP	60	[TCP Keep-Alive] 49756 → 445 [ACK] Seq=0 Ack=2 Win=8
4	0.192869	192.168.3.25	192.168.5.30	TCP	66	[TCP Keep-Alive ACK] 445 → 49756 [ACK] Seq=2 Ack=1 W
5	0.286727	192.168.5.30	192.168.3.25	SMB2	178	Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
6	0.287048	192.168.3.25	192.168.5.30	SMB2	474	Ioctl Response FSCTL_QUERY_NETWORK_INTERFACE_INFO
7	0.334283	192.168.5.30	192.168.3.25	TCP	60	49756 → 445 [ACK] Seq=125 Ack=422 Win=8209 Len=0
8	0.465952	192.168.3.25	192.168.6.25	TLSv1.2	102	Application Data
9	0.466564	192.168.3.25	192.168.6.25	TLSv1.2	102	Application Data
10	0.466842	192.168.6.25	192.168.3.25	TLSv1.2	1383	Application Data
11	0.466893	192.168.3.25	192.168.6.25	TCP	54	49868 → 443 [ACK] Seq=97 Ack=1330 Win=32768 Len=0
12	0.467335	192.168.6.25	192.168.3.25	TLSv1.2	1360	Application Data
13	0.467384	192.168.3.25	192.168.6.25	TCP	54	49868 → 443 [ACK] Seq=97 Ack=2636 Win=32604 Len=0
14	18.345052	192.168.3.25	192.168.6.25	TLSv1.2	152	Application Data
15	18.345809	192.168.3.25	192.168.6.25	TLSv1.2	162	Application Data
16	18.345936	192.168.3.25	192.168.6.25	TLSv1.2	92	Application Data
17	18.346304	192.168.6.25	192.168.3.25	TCP	60	443 → 49868 [ACK] Seq=2636 Ack=341 Win=2048 Len=0
18	18.349289	192.168.6.25	192.168.3.25	TLSv1.2	109	Application Data
19	18.349341	192.168.3.25	192.168.6.25	TCP	54	49868 → 443 [ACK] Seq=341 Ack=2691 Win=32597 Len=0
20	18.400732	192.168.6.25	192.168.3.25	TCP	66	49759 → 3306 [SYN, ECN, ChR] Seq=0 Win=8192 Len=0 MS
21	18.400816	192.168.3.25	192.168.6.25	TCP	66	3306 → 49759 [SYN, ACK, ECN] Seq=0 Ack=1 Win=8192 Le
22	18.401244	192.168.6.25	192.168.3.25	TCP	60	49759 → 3306 [ACK] Seq=1 Ack=1 Win=525568 Len=0
23	18.402607	192.168.3.25	192.168.6.25	MySQL	132	Server Greeting proto=10 version=8.0.25
24	18.403129	192.168.6.25	192.168.3.25	MySQL	208	Login Request user=root db=user_info
25	18.403383	192.168.3.25	192.168.6.25	MySQL	102	Auth Switch Request
26	18.403772	192.168.6.25	192.168.3.25	MySQL	78	Auth Switch Response

Directed to mo.11 is the beginning of when we are accessing the site. You can see that I am sending a TCP packet that has a ACK flag to the port 443. The port 443 is for sites with an ssl certificate, because I built the site with ssl the ports we are looking for is 443 as opposed to port 80 (which is more unsecure).

## SC#2 – Screenshot of the form being submitted to the fileserver

The screenshot displays a network traffic analysis tool. The top pane shows a list of packets. Packet 20 is highlighted, showing a MySQL query. The bottom pane provides a detailed view of this query, showing the SQL statement: `INSERT INTO user_info(name, email, comment) VALUES('amy', 'amy@gmail.com', 'MESSAGES EVERYWHERE');`

No.	Time	Source	Destination	Protocol	Length	Info
19	18.349341	192.168.3.25	192.168.6.25	TCP	54	49868 → 443 [ACK] Seq=341 Ack=2691 Win=32597 Len=0
20	18.400732	192.168.6.25	192.168.3.25	TCP	66	49759 → 3306 [SYN, ECH, CWR] Seq=0 Win=0 Len=0
21	18.400816	192.168.3.25	192.168.6.25	TCP	66	3306 → 49759 [SYN, ACK, ECH] Seq=0 Ack=1 Win=0 Len=0
22	18.401244	192.168.6.25	192.168.3.25	TCP	60	49759 → 3306 [ACK] Seq=1 Ack=1 Win=525568 Len=0
23	18.402607	192.168.6.25	192.168.3.25	MySQL	132	Server Greeting proto=10 version=8.0.25
24	18.403129	192.168.6.25	192.168.3.25	MySQL	208	Login Request user=root db=user_info
25	18.403383	192.168.3.25	192.168.6.25	MySQL	102	Auth Switch Request
26	18.403772	192.168.6.25	192.168.3.25	MySQL	78	Auth Switch Response
27	18.404494	192.168.3.25	192.168.6.25	MySQL	65	Response OK
28	18.404990	192.168.6.25	192.168.3.25	MySQL	118	Request Prepare Statement
29	18.406227	192.168.3.25	192.168.6.25	MySQL	168	Response
30	18.406711	192.168.6.25	192.168.3.25	MySQL	114	Request Execute Statement
31	18.410378	192.168.3.25	192.168.6.25	MySQL	65	Response OK
32	18.410887	192.168.6.25	192.168.3.25	MySQL	63	Request Close Statement
33	18.410887	192.168.6.25	192.168.3.25	MySQL	60	Request Quit
34	18.410978	192.168.3.25	192.168.6.25	TCP	54	3306 → 49759 [ACK] Seq=255 Ack=317 Win=525056 Len=0
35	18.411039	192.168.6.25	192.168.3.25	TCP	60	49759 → 3306 [FIN, ACK] Seq=317 Ack=255 Win=525312 Len=0
36	18.411042	192.168.3.25	192.168.6.25	TCP	54	3306 → 49759 [FIN, ACK] Seq=255 Ack=317 Win=525056 Len=0
37	18.411084	192.168.3.25	192.168.6.25	MySQL	54	3306 → 49759 [ACK] Seq=256 Ack=318 Win=525056 Len=0
38	18.411294	192.168.6.25	192.168.3.25	TCP	60	49759 → 3306 [ACK] Seq=318 Ack=256 Win=525312 Len=0
39	18.413644	192.168.6.25	192.168.3.25	TLSv1.2	211	Application Data
40	18.413721	192.168.3.25	192.168.6.25	TCP	54	49868 → 443 [ACK] Seq=341 Ack=2848 Win=32768 Len=0
41	22.311312	192.168.6.25	192.168.3.25	TLSv1.2	122	Application Data
42	22.311569	192.168.3.25	192.168.6.25	TLSv1.2	117	Application Data
43	22.311779	192.168.6.25	192.168.3.25	TLSv1.2	92	Application Data
44	22.311862	192.168.6.25	192.168.3.25	TCP	60	443 → 49868 [ACK] Seq=2848 Ack=472 Win=2047 Len=0

Internet Protocol Version 4, Src: 192.168.6.25, Dst: 192.168.3.25  
0100 .... = Version: 4  
... 0101 = Header Length: 20 bytes (5)  
> Differentiated Services Field: 0x02 (DSCP: CS0, ECN: ECT(0))  
Total Length: 52  
Identification: 0x2c33 (11315)  
> Flags: 0x40, Don't fragment  
Fragment Offset: 0  
Time to Live: 127  
Protocol: TCP (6)  
Header Checksum: 0x450c [validation disabled]  
[Header checksum status: Unverified]  
Source Address: 192.168.6.25

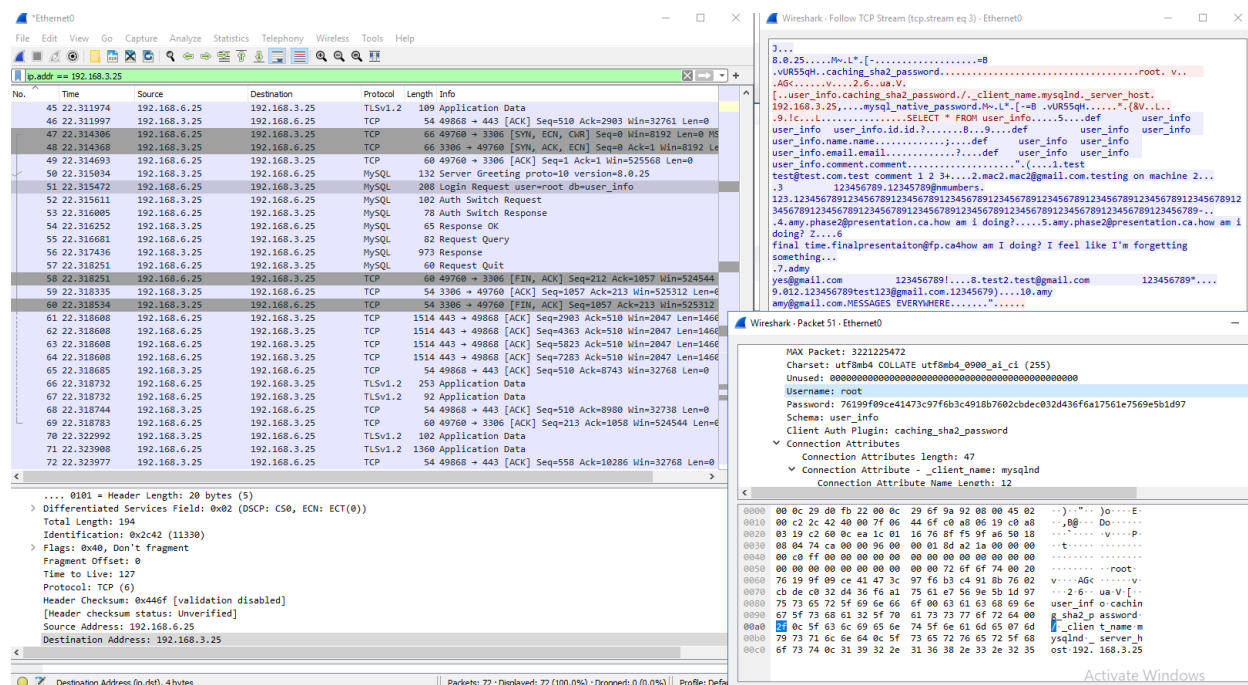
No.20 is when the form is being submitted to the database, we can see that the webserver is accessing the fileserver. When following the data stream we can see that there is information being inserted into the table. The data is being inserted into the table `user_info`, in the order of name, email, comment. In particular the “amy amy@gmail.com.MESSAGES EVERYWHERE” is indicative that it is being sent to the database.

From the screenshot we can conclude that the order of values being sent to the `use_info` table is:

User_info	
name	Amy
email	<a href="mailto:amy@gmail.com">amy@gmail.com</a>
comment	MESSAGES EVERYWHERE

No.36 is when we see that the handshake is finished because of the FIN flag.

### SC#3 – Screenshot of the form values being retrieved from the fileserver



The above Screenshot shows the form values being retrieved from the database. when following the TCP stream for packet 47 we can see that we SQL code I particular we can see `SELECT * FROM user_info`, this means that from the table `user_info` select the following data. The following data is defined by `user_info.id.id`, `user_info.name.name`, `user_info.email.email`, and `user_info.comment.comment`. We can then see all of the values entered using the website form.

This could be potentially problematic because we can get the MySQL username in this case which is root which can be seen in packet 51. The password is encrypted using SHA2, but it can be cracked if we compare hash strings of known passwords. In an enterprise it would be better to have a more complex password.

Being able to see the data also means that individuals can steal private and personal data. For example, if the table was instead `user_info.password.password`, `user_info.address.address`, `user_info.dob.dob`, etc. then this means someone can potentially steal data and use sql injections.

## B. Accessing document on fileserver

As background I am accessing the fileserver (192.168.3.25) using the adminpc (192.168.5.30) as Annie Hanson

### SC#4 – screenshot of the adminpc accessing a file on the file server

The screenshot displays two windows from a Windows operating system. The primary window is Wireshark, showing a network capture on the 'eth0' interface. The packet list pane on the left shows 28 packets. The packet details pane on the right shows the selected packet (No. 23) as a 'Close Request File: ip.txt' (SMB2). The packet bytes pane at the bottom shows the raw data of the packet, including the SMB header and the file path '\\filesaver\IPC\$'. A secondary window, 'Downloads', is open, showing a list of files. The file '192.168.1.1' is selected, and its content is displayed in a Notepad window. The content of the file is 'THIS IS FOR THE FP PDF'.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.5.30	192.168.3.25	SMB2	206	Create Request File:
2	0.000196	192.168.3.25	192.168.5.30	SMB2	322	Create Response File:
3	0.027555	192.168.5.30	192.168.3.25	SMB2	146	Close Request File:
4	0.027698	192.168.3.25	192.168.5.30	SMB2	182	Close Response
5	0.082133	192.168.5.30	192.168.3.25	TCP	60	49903 → 445 [ACK] Seq=325 Ack=397 Win=8212 Len=0
6	1.508819	192.168.5.30	192.168.3.25	SMB2	274	Create Request File: ip.txt:Zone.Identifier
7	1.508940	192.168.3.25	192.168.5.30	SMB2	131	Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND
8	1.660913	192.168.5.30	192.168.3.25	TCP	60	49903 → 445 [ACK] Seq=545 Ack=474 Win=8211 Len=0
9	1.684625	192.168.5.30	192.168.3.25	SMB2	164	Tree Connect Request Tree: \\filesaver\IPC\$
10	1.684742	192.168.3.25	192.168.5.30	SMB2	138	Tree Connect Response
11	1.700564	192.168.5.30	192.168.3.25	SMB2	190	Create Request File: srvsvc
12	1.700930	192.168.3.25	192.168.5.30	SMB2	210	Create Response File: srvsvc
13	1.701412	192.168.5.30	192.168.3.25	SMB2	162	GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO FI
14	1.701483	192.168.3.25	192.168.5.30	SMB2	154	GetInfo Response
15	1.702021	192.168.5.30	192.168.3.25	DCERPC	330	Bind: call_id: 2, Fragment: Single, 3 context items:
16	1.702085	192.168.3.25	192.168.5.30	SMB2	138	Write Response
17	1.702407	192.168.5.30	192.168.3.25	SMB2	171	Read Request Len:1024 Off:0 File: srvsvc
18	1.702452	192.168.3.25	192.168.5.30	DCERPC	254	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 42
19	1.702756	192.168.5.30	192.168.3.25	SRVSVC	298	NetShareGetInfo request
20	1.702949	192.168.3.25	192.168.5.30	SRVSVC	306	NetShareGetInfo response
21	1.703451	192.168.5.30	192.168.3.25	SMB2	146	Close Request File: srvsvc
22	1.703506	192.168.3.25	192.168.5.30	SMB2	182	Close Response
23	1.706759	192.168.5.30	192.168.3.25	SMB2	362	Create Request File: ip.txt:ioctl Request FSCTL_CREA
24	1.706983	192.168.3.25	192.168.5.30	SMB2	474	Create Response File: ip.txt:ioctl Response FSCTL_C
25	1.707501	192.168.5.30	192.168.3.25	SMB2	162	GetInfo Request FS_INFO/FilesObjectInformation FI
26	1.707677	192.168.3.25	192.168.5.30	SMB2	194	GetInfo Response
27	1.709414	192.168.5.30	192.168.3.25	SMB2	146	Close Request File: ip.txt
28	1.710052	192.168.3.25	192.168.5.30	SMB2	182	Close Response

.... 0101 = Header Length: 20 bytes (5)  
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
Total Length: 140  
Identification: 0xc4e1 (50401)  
Flags: 0x40, Don't fragment  
Fragment Offset: 0  
Time to Live: 127  
Protocol: TCP (6)  
Header Checksum: 0xacfa (validation disabled)  
[Header checksum status: Unverified]  
Source Address: 192.168.5.30  
Destination Address: 192.168.3.25

Packet	Hostname	Content Type	Size	Filename
15	\\filesaver\IPC\$	FILE (160/160) R&W (100.00%)	160 bytes	\\srvsvc
49	\\192.168.3.25\TREEID_UNKNOWN	FILE (25/25) R (100.00%)	25 bytes	\\ip.txt
77	\\filesaver\IPC\$	FILE (160/160) R&W (100.00%)	160 bytes	\\srvsvc
111	\\192.168.3.25\TREEID_UNKNOWN	FILE (64/64) W (100.00%)	64 bytes	\\ip.txt

Above we can see an abundance of information. If we follow the streams of any of the packets most of the data that we can see is ineligible for humans. However we can see that the adminpc creates a request file, the request files and its response is what gives the adminpc allowance to work in those files requested. We can also see the server of what we are trying to reach which is \\filesaver\IPC\$. we can also see that there is a request file for ip.txt in packet no.23 what this means is that adminpc is trying to get access to ip.txt. So the file server sends a GetInfo Request to verify if the adminpc has appropriate permissions. When the GetInfo Response is finished the request and response is then closed, and now the adminpc has access to the file.

We can also export files, in the example above we exported all of the files. When opening

%5cip(1) we can see the new text added which is:

“192.168.10.30

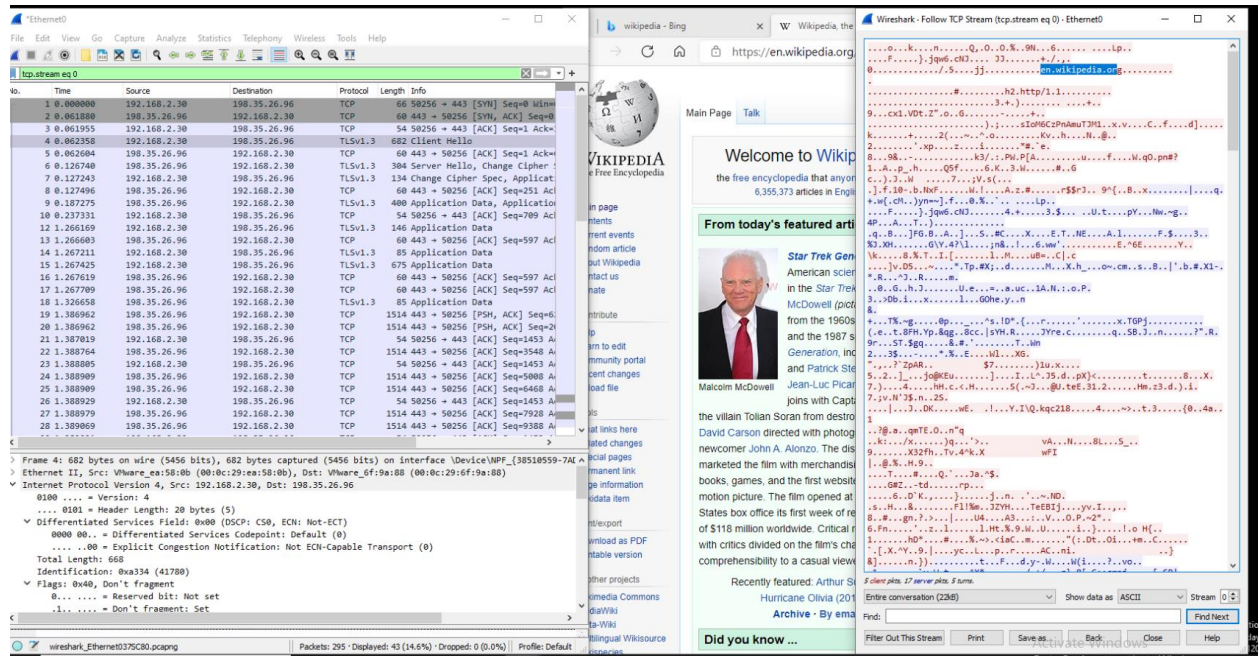
This is for the FP PDF”

What this means is that we can see the file before edits and after edits.

### C. Accessing the internet website

To access the internet website I am going to use pcb to access the site [www.youtube.com](http://www.youtube.com)

**SC#5 – screenshot of pcb accessing the internet site wikipedia.com**



Packet no.3 shows pcb accessing the Wikipedia site. When packet no.3 is followed we can see that en.wikipedia.com is being accessed. The reason for the sizable pcap file is because of the loading of resources. The pcap file has 295 packets, and while the site was loading it needed to access the images, links, etc. which took longer than usual.