

#31. Wireshark로 패킷 Capture하기

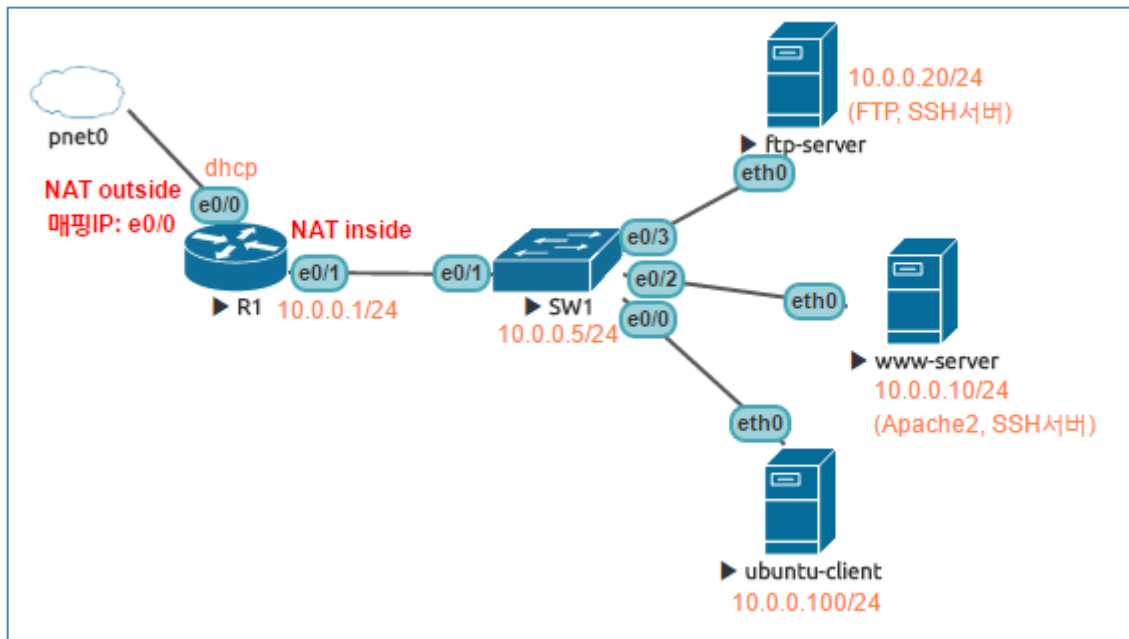
문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

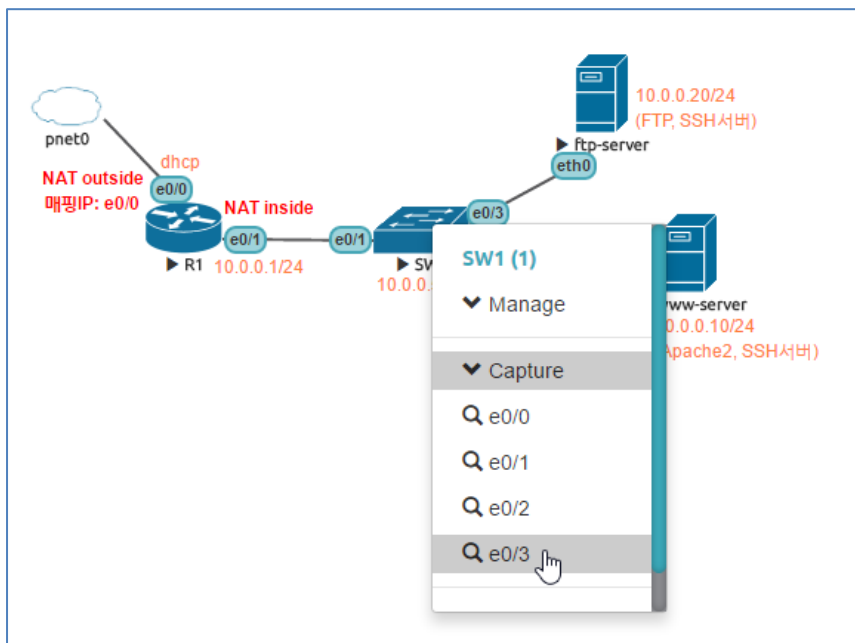
이번 LAB은 “#30. Docker시스템으로 서버구현하기”에서 구성한 LAB을 이용하여 클라이언트와 서버간에 TCP통신이 수행될 때 Wireshark로 패킷을 잡아보고 패킷을 분석해보는 LAB입니다. 패킷을 ASCII형태로 decode했을 때 어떻게 보이는지와 Flow Graph에서 서버와 클라이언트가 사용하는 TCP port번호를 주의깊게 살펴보세요.

[LAB구성도]



1. Capture실행방법

- ① UnetLab에서 capture를 수행하기 위해서는 “#24. UnetLab에서 WireShark로 패킷캡처”LAB의 설정이 완료되어야 합니다.
- ② LAB구성도에서 SW1에 오른쪽버튼을 클릭하고 capture메뉴를 클릭하여 capture를 수행할 포트를 선택하면 wireshark를 구동해주는 batch파일이 구동되어서 Wireshark가 정상적으로 구동되어야 합니다.

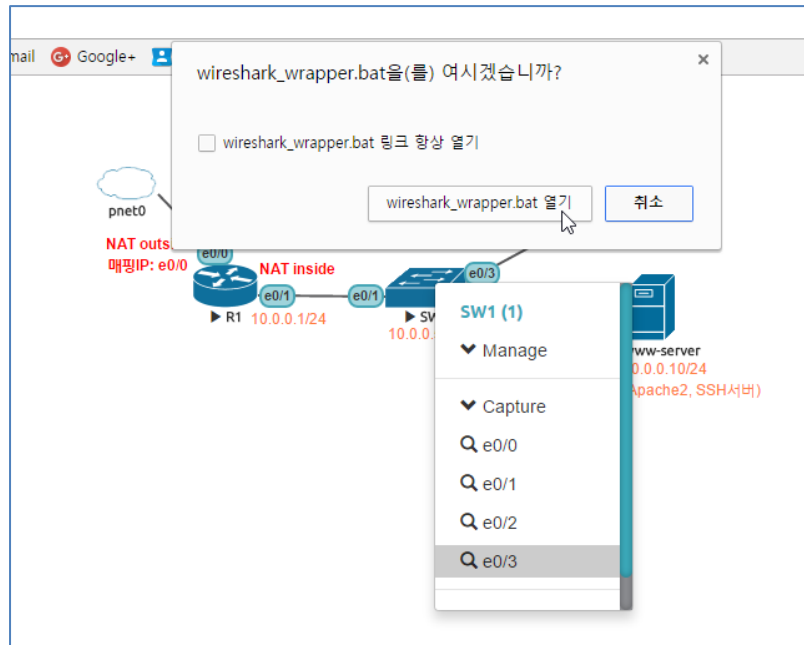


#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



- ③ Analyze를 수행한 후에는 패킷이 필터링되어 있는 상태이므로 필터부분의 X버튼으로 clear해서 capture한 패킷이 모두 보이도록 한 후에 다음 분석을 진행합니다.

*Standard input

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

Expression...

No.	Time	Source	Destination	Protocol	Length	Info
8	9.318728	10.0.0.100	10.0.0.10	TCP	74	39330→80 [SYN] Seq=0 Win=29200 Len=0 M...
9	9.318838	10.0.0.10	10.0.0.100	TCP	74	80→39330 [SYN, ACK] Seq=0 Ack=1 Win=28...
10	9.319593	10.0.0.100	10.0.0.10	TCP	66	39330→80 [ACK] Seq=1 Ack=1 Win=29312 L...
11	9.320779	10.0.0.100	10.0.0.10	HTTP	319	GET / HTTP/1.0
12	9.320976	10.0.0.10	10.0.0.100	TCP	66	80→39330 [ACK] Seq=1 Ack=254 Win=30080...
13	9.339921	10.0.0.10	10.0.0.100	TCP	1514	[TCP segment of a reassembled PDU]
14	9.339940	10.0.0.10	10.0.0.100	TCP	1514	[TCP segment of a reassembled PDU]
15	9.339975	10.0.0.10	10.0.0.100	HTTP	658	HTTP/1.1 200 OK (text/html)
16	9.341938	10.0.0.100	10.0.0.10	TCP	66	39330→80 [ACK] Seq=254 Ack=1449 Win=32...
17	9.342080	10.0.0.100	10.0.0.10	TCP	66	39330→80 [ACK] Seq=254 Ack=2897 Win=35...
18	9.342176	10.0.0.100	10.0.0.10	TCP	66	39330→80 [ACK] Seq=254 Ack=3489 Win=37...
19	9.354305	10.0.0.10	10.0.0.100	TCP	66	80→39330 [FIN, ACK] Seq=3489 Ack=254 W...
20	9.371297	10.0.0.100	10.0.0.10	TCP	66	39330→80 [FIN, ACK] Seq=254 Ack=3490 W...
21	9.371387	10.0.0.10	10.0.0.100	TCP	66	80→39330 [ACK] Seq=3490 Ack=255 Win=30...

> Frame 8: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: 50:00:00:04:00:00 (50:00:00:04:00:00)

> Internet Protocol Version 4, Src: 10.0.0.100, Dst: 10.0.0.10

> Transmission Control Protocol, Src Port: 39330, Dst Port: 80, Seq: 0, Len: 0

```

0000  50 00 00 04 00 00 50 00 00 02 00 00 08 00 45 00  P....P. ....E.
0010  00 3c c9 dc 40 00 40 06 5c 72 0a 00 00 64 0a 00  .<..@. \r...d..
0020  00 0a 99 a2 00 50 14 d7 89 4f 00 00 00 00 a0 02  ....P.. .0.....
0030  72 10 48 d2 00 00 02 04 05 b4 04 02 08 0a 00 04  r.H.....
0040  40 93 00 00 00 00 01 03 03 07                      @.....
    
```

wireshark_-_20170202095413_a15832

Packets: 82 · Displayed: 14 (17.1%)

Profile: Default

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

2. ubuntu-client(10.0.0.100)에서 SW1(10.0.0.5)로 telnet,ssh연결에 대한 Capture

이 capture에서는 동일 시스템을 2가지 방법(telnet, ssh)으로 원격연결을 수행하면서 수행합니다. 두 가지 방법에 대한 ASCII 분석을 살펴보면 telnet의 경우에 사용자가 입력하는 계정, 암호와 명령들이 모두 보이는 것을 확인할 수 있고, ssh를 사용하는 경우에는 암호화되어서 보이지 않는다는 것을 확인할 수 있습니다. 장비들 설정에서 telnet을 사용하지 말고 ssh로 변경하라고 하는지 이해가 되시나요?

- ① SW1스위치에 telnet, ssh연결이 가능하도록 설정완료할 것
- ② SW1스위치의 e0/0포트에 대한 capture를 시작
- ③ ubuntu-client에서 telnet 10.0.0.5로 접속하고 명령수행후에 연결종료. Wireshark의 capture중지
- ④ Wireshark에서 capture패킷중에 첫번째 TCP패킷을 클릭한 다음에 “Analyze > Follow > TCP Stream”선택

The screenshot shows the Wireshark interface with a packet capture of a telnet session. The packet list shows a TCP SYN packet (74 bytes) from 10.0.0.100 to 10.0.0.5. The packet details pane shows the TCP stream information. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.100	10.0.0.5	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:...
2	2.009777	10.0.0.100	10.0.0.5	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:...
3	4.018874	10.0.0.100	10.0.0.5	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:...
4	6.024806	10.0.0.100	10.0.0.5	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:...
5	7.250661	10.0.0.100	10.0.0.5	ARP	42	Who has 10.0.0.5? Tell 10.0.0.100
6	7.251126	10.0.0.100	10.0.0.5	ARP	60	10.0.0.5 is at aa:bb:cc:80:01:00
7	7.251159	10.0.0.100	10.0.0.5	TCP	74	57219→23 [SYN] Seq=0 Win=29200 Le...
8	7.251354	10.0.0.100	10.0.0.5	ARP	60	Who has 10.0.0.100? Tell 10.0.0.5
9	7.251366	10.0.0.100	10.0.0.5	ARP	42	10.0.0.100 is at 50:00:00:02:00:00
10	8.029884	10.0.0.100	10.0.0.5	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:...
11	8.248292	10.0.0.100	10.0.0.5	TCP	74	[TCP Retransmission] 57219→23 [SY...
12	8.249254	10.0.0.100	10.0.0.5	TCP	60	23→57219 [ACK] Seq=1 Ack=1 Win=41...
13	9.259401	10.0.0.100	10.0.0.5	TCP	60	[TCP Port numbers reused] 23→5721...
14	9.259577	10.0.0.100	10.0.0.5	TCP	54	57219→23 [ACK] Seq=1 Ack=2 Win=29...
15	9.260504	10.0.0.100	10.0.0.5	TELNET	78	Telnet Data ...
16	9.270515	10.0.0.100	10.0.0.5	TELNET	66	Telnet Data ...

Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: aa:bb:cc:80:01:00 (aa:bb:cc:80:01:00)
 Internet Protocol Version 4, Src: 10.0.0.100, Dst: 10.0.0.5
 Transmission Control Protocol, Src Port: 57219, Dst Port: 23, Seq: 0, Len: 0

0000 aa bb cc 80 01 00 50 00 00 02 00 00 08 00 45 10P.E.
 0010 00 3c 2e a7 40 00 40 06 f7 9c 0a 00 00 64 0a 00 .<..@.@.d..
 0020 00 05 df 83 00 17 93 61 26 e1 00 00 00 00 a0 02a &.....
 0030 72 10 1f 5f 00 00 02 04 05 b4 04 02 08 0a 00 05 r.....
 0040 08 46 00 00 00 00 01 03 03 07 .F.....

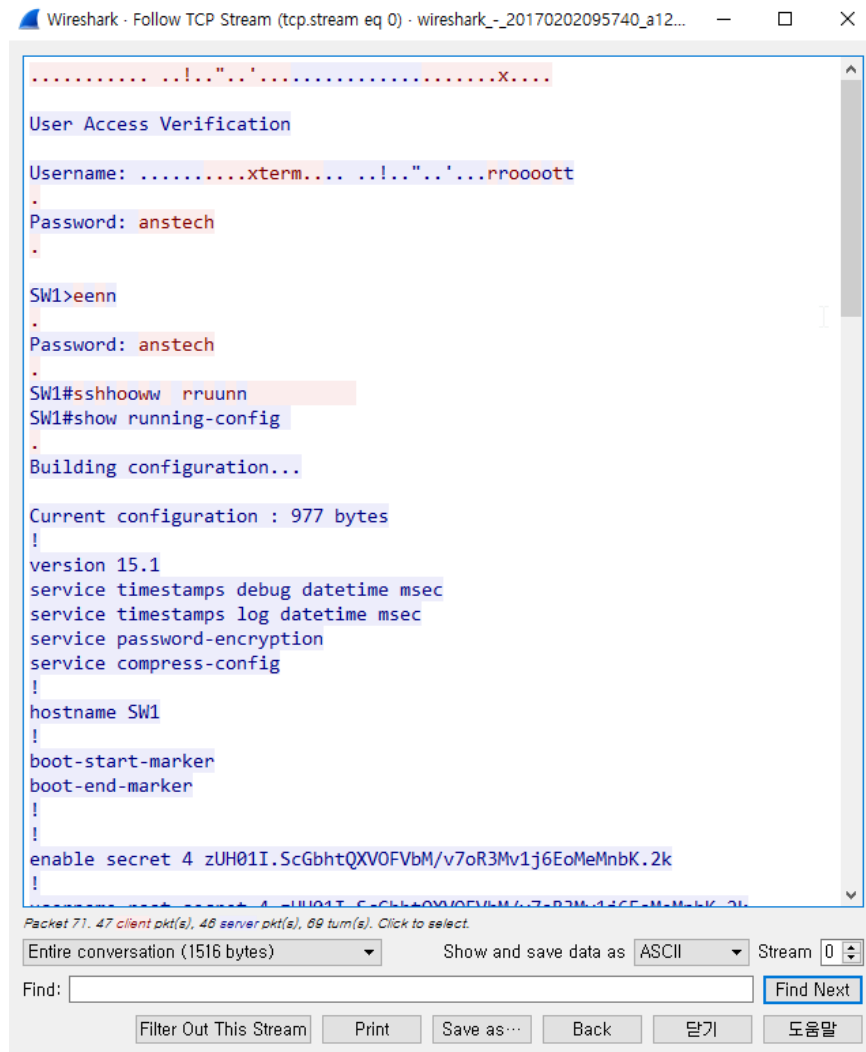
- ⑤ 아래와 같이 ASCII형태로 telnet packet에 대한 분석을 보여줌.

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



```

.....X....
User Access Verification
Username: .....xterm.....!.."'.rroooott
Password: ansstech
SW1>eenn
Password: ansstech
SW1#sshhooww rruunn
SW1#show running-config
Building configuration...

Current configuration : 977 bytes
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service compress-config
!
hostname SW1
!
boot-start-marker
boot-end-marker
!
enable secret 4 zUH01I.ScGbhtQXV0FVbM/v7oR3Mv1j6EoMeMnbK.2k
!

```

Packet 71: 47 client pkt(s), 46 server pkt(s), 69 turn(s). Click to select.

Entire conversation (1516 bytes) Show and save data as ASCII Stream 0

Find: Find Next

Filter Out This Stream Print Save as... Back 닫기 도움말

- ⑥ 필터링부분을 clear한 후에 “Statistics > Flow Graph”를 선택하고 “TCP Flow”를 선택하면 아래와 같이 TCP Flow에 대한 그래프가 나옴. 그래프에서 클라이언트와 서버측의 통신포트를 확인해보기

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

Wireshark - Flow - wireshark_-_20170202095740_a12064

Time	10.0.0.100	10.0.0.5	Comment
7.251159	57219	23	Seq = 0
8.248292	57219	23	Seq = 0
8.249254	57219	23	Seq = 1 Ack = 1
9.259401	57219	23	Seq = 1 Ack = 1
9.259577	57219	23	Seq = 1 Ack = 2
9.260504	57219	23	Seq = 1 Ack = 2
9.270515	57219	23	Seq = 2 Ack = 25
9.270658	57219	23	Seq = 25 Ack = 14
9.270806	57219	23	Seq = 25 Ack = 14
9.278757	57219	23	Seq = 14 Ack = 37
9.284522	57219	23	Seq = 56 Ack = 37
9.284711	57219	23	Seq = 37 Ack = 62
9.284822	57219	23	Seq = 37 Ack = 62
9.285157	57219	23	Seq = 62 Ack = 37
9.285415	57219	23	Seq = 65 Ack = 37
9.285514	57219	23	Seq = 48 Ack = 68
9.285697	57219	23	Seq = 68 Ack = 37
9.286154	57219	23	Seq = 71 Ack = 37
9.286351	57219	23	Seq = 48 Ack = 74
9.286550	57219	23	Seq = 74 Ack = 37
9.325100	57219	23	Seq = 48 Ack = 77
9.496696	57219	23	Seq = 37 Ack = 77
9.497189	57219	23	Seq = 77 Ack = 48

2 node(s), 152 item(s)

Show: All packets

Flow type: TCP Flows

Addresses: Any

Save As... 닫기 도움말

- ⑦ Wireshark를 종료하고 다시 SW1스위치의 e0/0포트에 대한 capture를 시작
- ⑧ ubuntu-client에서 ssh root@10.0.0.5로 접속하고 명령수행후에 연결종료. Wireshark의 capture중지
- ⑨ Wireshark에서 capture패킷중에 첫번째 TCP패킷을 클릭한 다음에 “Analyze > Follow > TCP Stream”선택

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

The screenshot shows the Wireshark interface with the following details:

- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	aa:bb:cc:80:01:00	50:00:00:02:00:00	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:00
2	0.990064	aa:bb:cc:80:01:00	10.0.0.5	TCP	74	47035→22 [SYN] Seq=0 Win=29200 Len=0
3	2.009740	aa:bb:cc:80:01:00	10.0.0.5	TCP	60	22→47035 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
4	4.018812	aa:bb:cc:80:01:00	10.0.0.5	TCP	54	47035→22 [ACK] Seq=1 Ack=1 Win=29200 Len=0
5	6.020697	aa:bb:cc:80:01:00	10.0.0.5	TCP	95	Client: Protocol (SSH-2.0-OpenSSH_7.3p1)
6	8.025015	aa:bb:cc:80:01:00	10.0.0.5	TCP	73	Server: Protocol (SSH-2.0-Cisco-1.2)
7	9.210763	10.0.0.100	10.0.0.5	SSHv2	54	47035→22 [ACK] Seq=42 Ack=20 Win=29200 Len=0
8	9.211253	10.0.0.100	10.0.0.5	SSHv2	1390	Client: Key Exchange Init
9	9.211298	10.0.0.100	10.0.0.5	SSHv2	398	Server: Key Exchange Init
10	9.211601	10.0.0.100	10.0.0.5	SSHv2	78	Client: Diffie-Hellman Group Exchange
- Packet Details:**
 - Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 - Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: aa:bb:cc:80:01:00 (aa:bb:cc:80:01:00)
 - Internet Protocol Version 4, Src: 10.0.0.100, Dst: 10.0.0.5
 - Transmission Control Protocol, Src Port: 47035, Dst Port: 22, Seq: 0, Len: 0
- Packet Bytes:**

```

0000 aa bb cc 80 01 00 50 00 00 02 00 00 08 00 45 00 .....P. ....E.
0010 00 3c f6 eb 40 00 40 06 2f 68 0a 00 00 64 0a 00 .<..@.@. /h...d..
0020 00 05 b7 bb 00 16 be c1 19 a6 00 00 00 00 a0 02 .....
0030 72 10 41 fb 00 00 02 04 05 b4 04 02 08 0a 00 05 r.A.....
0040 ef 4d 00 00 00 00 01 03 03 07 .M.....

```

⑩ 아래와 같이 ASCII형태로 telnet packet에 대한 분석을 보여줌.

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



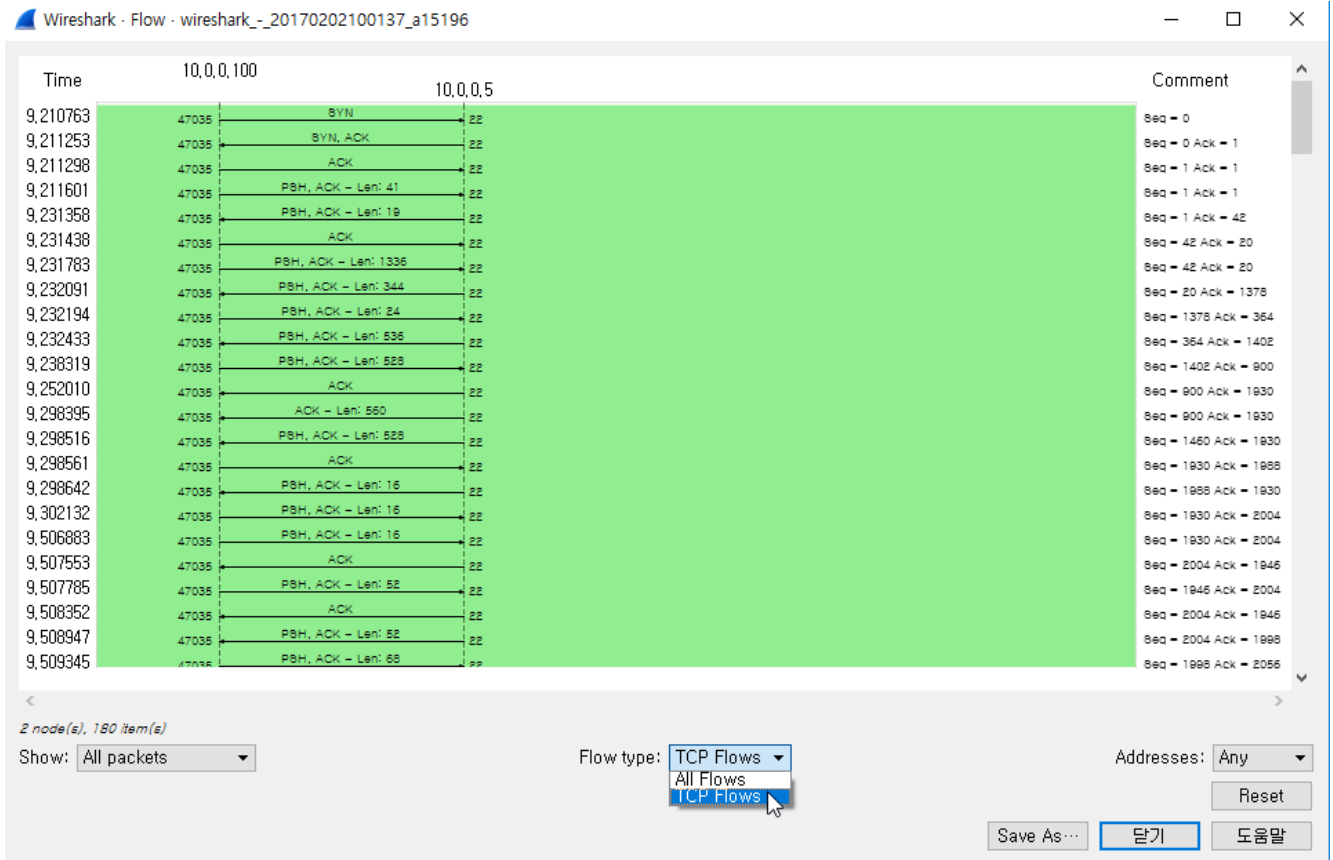
- ⑪ 필터링부분을 clear한 후에 “Statistics > Flow Graph”를 선택하고 “TCP Flow”를 선택하면 아래와 같이 TCP Flow에 대한 그래프가 나옴. 그래프에서 클라이언트와 서버측의 통신포트를 확인해보기

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



3. ubuntu-client(10.0.0.100)에서 ftp-server(10.0.0.20)로 ftp, sftp연결에 대한 Capture

이 capture에서는 파일전송을 위해서 동일 시스템을 2가지 방법(ftp, sftp)으로 연결을 수행하면서 수행합니다. 두 가지 방법에 대한 ASCII 분석을 살펴보면 ftp의 경우에 사용자가 입력하는 계정, 암호와 명령들이 모두 보이는 것을 확인할 수 있고, sftp를 사용하는 경우에는 암호화되어서 보이지 않는다는 것을 확인할 수 있습니다. 파일전송할때 ftp를 사용하지 말아야하는 이유가 이해되시나요?

- ① ftp-server에 vsftpd데몬과 ssh데몬을 구동할 것
- ② SW1스위치의 e0/3포트에 대한 capture를 시작
- ③ ubuntu-client에서 ftp 10.0.0.20으로 접속하고 명령수행후에 연결종료. Wireshark의 capture중지
- ④ Wireshark에서 capture패킷중에 첫번째 TCP패킷을 클릭한 다음에 “Analyze > Follow > TCP Stream”선택

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

The screenshot shows the Wireshark interface with the following details:

- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
11	14.505264	50:00:00:02:00:00	50:00:00:05:00:00	Broadcast	60	Conf. Root = 32768/1/aa:bb:cc:00:...
12	16.045702	10.0.0.100	10.0.0.20	TCP	74	39272→21 [SYN] Seq=0 Win=29200 Le...
13	18.049071	10.0.0.20	10.0.0.100	TCP	74	21→39272 [SYN, ACK] Seq=0 Ack=1 W...
- Packet Details:**
 - Frame 16: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 - Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: 50:00:00:05:00:00 (50:00:00:05:00:00)
 - Internet Protocol Version 4, Src: 10.0.0.100, Dst: 10.0.0.20
 - Transmission Control Protocol, Src Port: 39272, Dst Port: 21, Seq: 0, Len: 0
- Packet Bytes:**

```

0000  50 00 00 05 00 00 50 00 00 02 00 00 08 00 45 00  P.....P. ....E.
0010  00 3c e7 77 40 00 40 06 3e cd 0a 00 00 64 0a 00  .<.w@.@. >....d..
0020  00 14 99 68 00 15 2e 46 fa 30 00 00 00 00 a0 02  ...h...F .0.....
0030  72 10 e2 3b 00 00 02 04 05 b4 04 02 08 0a 00 01  r.;.... ....
0040  1d 47 00 00 00 00 01 03 03 07                    .G.....
    
```

- ⑤ 아래와 같이 ASCII형태로 ftp packet에 대한 분석을 보여줌.

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



```
220 (vsFTPD 3.0.3)
USER admin
331 Please specify the password.
PASS unl
230 Login successful.
SYST
215 UNIX Type: L8
PORT 10,0,0,100,151,14
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
PASV
227 Entering Passive Mode (10,0,0,20,222,222).
LIST
150 Here comes the directory listing.
226 Directory send OK.
CWD ..
250 Directory successfully changed.
PASV
227 Entering Passive Mode (10,0,0,20,106,192).
LIST
150 Here comes the directory listing.
226 Directory send OK.
CWD admin
250 Directory successfully changed.
CWD ..
250 Directory successfully changed.
CWD ..
250 Directory successfully changed.
PASV
227 Entering Passive Mode (10,0,0,20,164,188).
LIST
150 Here comes the directory listing.
226 Directory send OK.
```

Packet 41: 19 client pkt(s), 22 server pkt(s), 35 turn(s). Click to select.

Entire conversation (840 bytes) Show and save data as ASCII Stream 0

Find: Find Next

Filter Out This Stream Print Save as... Back 닫기 도움말

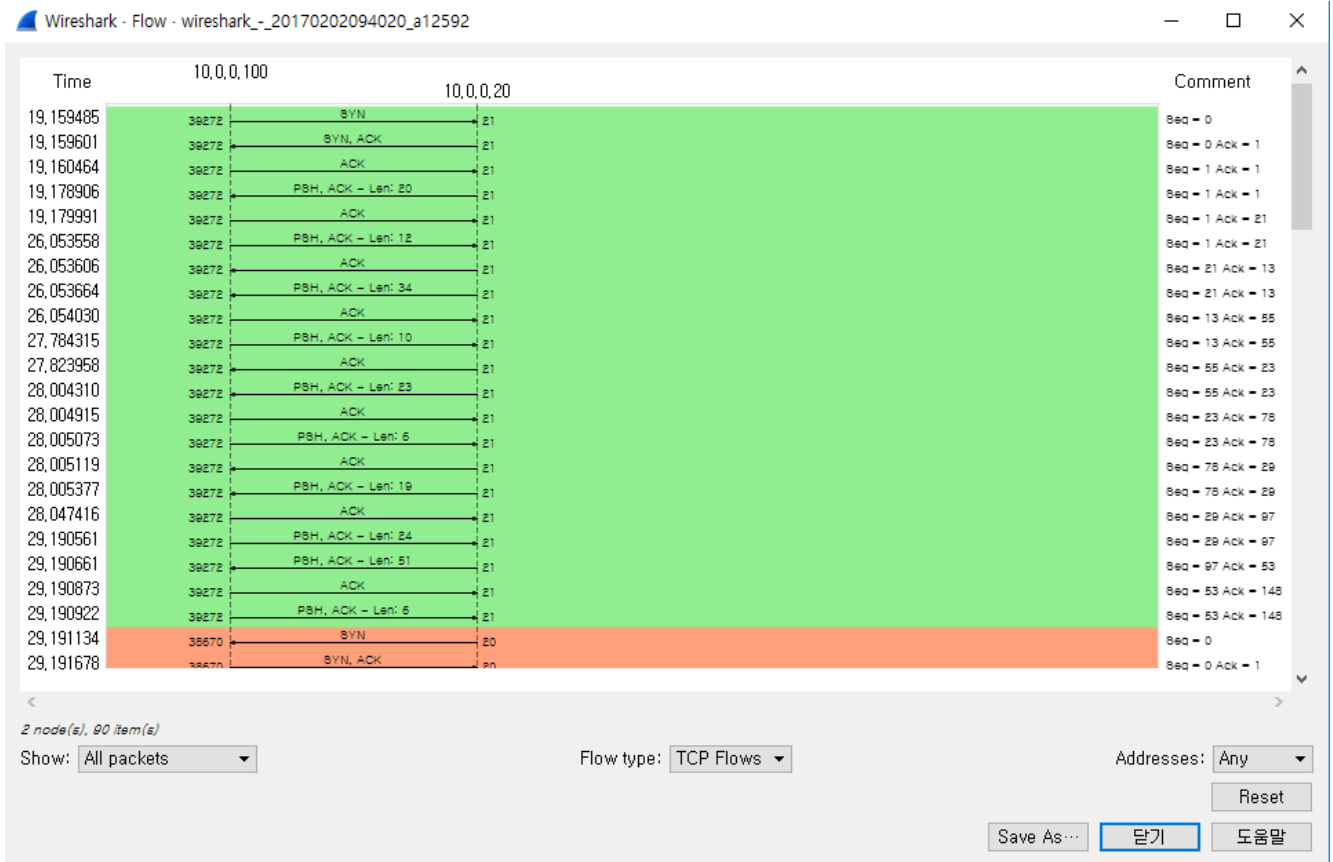
- ⑥ 필터링부분을 clear한 후에 “Statistics > Flow Graph”를 선택하고 “TCP Flow”를 선택하면 아래와 같이 TCP Flow에 대한 그래프가 나옴. 그래프에서 클라이언트와 서버측의 통신포트를 확인해보기

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



- ⑦ Wireshark를 종료하고 다시 SW1스위치의 e0/3포트에 대한 capture를 시작
- ⑧ ubuntu-client에서 sftp root@10.0.0.20으로 접속하고 명령수행후에 연결종료. Wireshark의 capture중지
- ⑨ Wireshark에서 capture패킷중에 첫번째 TCP패킷을 클릭한 다음에 “Analyze > Follow > TCP Stream”선택

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

The screenshot shows the Wireshark interface with the following details:

- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
6	9.470358	10.0.0.100	10.0.0.20	TCP	74	59135→22 [SYN] Seq=0 Win=29200 Len=0
- Packet Details:**
 - Frame 6: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 - Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: 50:00:00:05:00:00 (50:00:00:05:00:00)
 - Internet Protocol Version 4, Src: 10.0.0.100, Dst: 10.0.0.20
 - Transmission Control Protocol, Src Port: 59135, Dst Port: 22, Seq: 0, Len: 0
- Packet Bytes:**

```

0000  50 00 00 05 00 00 50 00 00 02 00 00 08 00 45 00  P....P. ....E.
0010  00 3c ad ba 40 00 40 06 78 8a 0a 00 00 64 0a 00  .<..@.@. x....d..
0020  00 14 e6 ff 00 16 75 5f 16 a5 00 00 00 00 a0 02  .....u .....
0030  72 10 93 38 00 00 02 04 05 b4 04 02 08 0a 00 06  r..8.....
0040  bb 1f 00 00 00 00 01 03 03 07  .....
    
```

⑩ 아래와 같이 ASCII형태로 sftp packet에 대한 분석을 보여줌.

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



- ⑪ 필터링부분을 clear한 후에 “Statistics > Flow Graph”를 선택하고 “TCP Flow”를 선택하면 아래와 같이 TCP Flow에 대한 그래프가 나옴. 그래프에서 클라이언트와 서버측의 통신포트를 확인해보기

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

Wireshark · Flow · wireshark_-_20170202100505_a10860

Time	10.0.0.100	10.0.0.20	Comment
9.470358	59135	22	Seq = 0
9.470412	59135	22	Seq = 0 Ack = 1
9.470979	59135	22	Seq = 1 Ack = 1
9.471504	59135	22	Seq = 1 Ack = 1
9.471546	59135	22	Seq = 1 Ack = 42
9.477791	59135	22	Seq = 1 Ack = 42
9.478735	59135	22	Seq = 42 Ack = 42
9.478769	59135	22	Seq = 42 Ack = 42
9.478826	59135	22	Seq = 42 Ack = 42
9.515685	59135	22	Seq = 1018 Ack = 1378
9.516281	59135	22	Seq = 1378 Ack = 1018
9.516499	59135	22	Seq = 1378 Ack = 1018
9.516561	59135	22	Seq = 1018 Ack = 1426
9.537428	59135	22	Seq = 1018 Ack = 1426
9.538272	59135	22	Seq = 1426 Ack = 1382
9.545006	59135	22	Seq = 1426 Ack = 1382
9.583193	59135	22	Seq = 1382 Ack = 1442
9.583531	59135	22	Seq = 1442 Ack = 1382
9.583567	59135	22	Seq = 1382 Ack = 1486
9.583821	59135	22	Seq = 1382 Ack = 1486
9.584143	59135	22	Seq = 1486 Ack = 1426
9.584961	59135	22	Seq = 1426 Ack = 1554
9.623568	59135	22	Seq = 1554 Ack = 1478

2 node(s), 96 item(s)

Show: All packets

Flow type: TCP Flows
All Flows
ICMP Flows

Addresses: Any

Save As... 닫기 도움말

4. ubuntu-client(10.0.0.100)에서 www-server(10.0.0.10)로 http연결에 대한 Capture

이 capture에서는 클라이언트에서 브라우저로 웹서버를 연결했을 때 패킷을 capture하는것으로 주고받는 HTML tag와 status code값들을 주의깊게 살펴보세요.

- ① SW1스위치의 e0/2포트에 대한 capture를 시작
- ② ubuntu-client에서 lynx http://10.0.0.10으로 웹서버 접속하고 시작페이지를 확인한 후에 q명령으로 lynx브라우저로 종료. Wireshark의 capture중지
- ③ Wireshark에서 capture패킷중에 첫번째 TCP패킷을 클릭한 다음에 “Analyze > Follow > TCP Stream”선택

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

Wireshark interface showing packet capture and analysis. The packet list shows a TCP SYN packet (Frame 8) from 10.0.0.10 to 10.0.0.100. The packet details pane shows the structure of the TCP segment. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.10	10.0.0.100	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:...
2	2.009804	10.0.0.10	10.0.0.100	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:...
3	4.018832	10.0.0.10	10.0.0.100	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:...
4	6.024316	10.0.0.10	10.0.0.100	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:...
5	8.024800	10.0.0.10	10.0.0.100	STP	60	Conf. Root = 32768/1/aa:bb:cc:00:01:...
6	9.317891	10.0.0.10	10.0.0.100	ARP	42	Who has 10.0.0.10? Tell 10.0.0.100
7	9.317964	10.0.0.10	10.0.0.100	ARP	42	10.0.0.10 is at 50:00:00:04:00:00
8	9.318728	10.0.0.10	10.0.0.100	TCP	74	39330→80 [SYN] Seq=0 Win=29200 Len=0...
9	9.318838	10.0.0.10	10.0.0.100	TCP	74	80→39330 [SYN, ACK] Seq=0 Ack=1 Win=...
10	9.319593	10.0.0.10	10.0.0.100	TCP	66	39330→80 [ACK] Seq=1 Ack=1 Win=29312...
11	9.320779	10.0.0.10	10.0.0.100	HTTP	319	GET / HTTP/1.0
12	9.320976	10.0.0.10	10.0.0.100	HTTP	66	80→39330 [ACK] Seq=1 Ack=254 Win=300...
13	9.339921	10.0.0.10	10.0.0.100	TCP	1514	[TCP segment of a reassembled PDU]
14	9.339940	10.0.0.10	10.0.0.100	TCP	1514	[TCP segment of a reassembled PDU]
15	9.339975	10.0.0.10	10.0.0.100	HTTP	658	HTTP/1.1 200 OK (text/html)
16	9.341938	10.0.0.10	10.0.0.100	TCP	66	39330→80 [ACK] Seq=254 Ack=1449 Win=...

Frame 8: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 > Ethernet II, Src: 50:00:00:02:00:00 (50:00:00:02:00:00), Dst: 50:00:00:04:00:00 (50:00:00:04:00:00)
 > Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.10
 > Transmission Control Protocol, Src Port: 39330, Dst Port: 80, Seq: 0, Len: 0

0000 50 00 00 04 00 00 50 00 00 02 00 00 08 00 45 00 P.....P.E.
 0010 00 3c c9 dc 40 00 40 06 5c 72 0a 00 00 64 0a 00 .<..@.@. \r...d..
 0020 00 0a 99 a2 00 50 14 d7 89 4f 00 00 00 00 a0 02P.. .O.....
 0030 72 10 48 d2 00 00 02 04 05 b4 04 02 08 0a 00 04 r.H.....
 0040 40 93 00 00 00 00 01 03 03 07 @.....

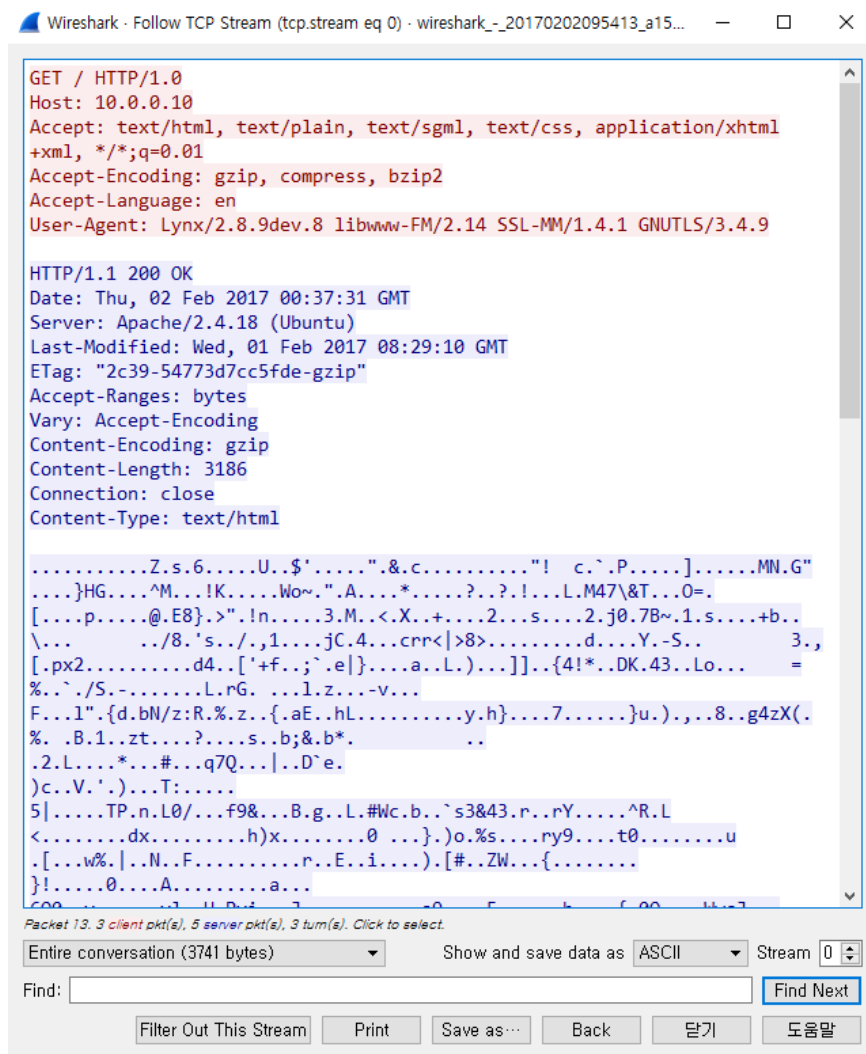
- ④ 아래와 같이 ASCII형태로 http packet에 대한 분석을 보여줌.

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02



- ⑤ 필터링부분을 clear한 후에 “Statistics > Flow Graph”를 선택하고 “TCP Flow”를 선택하면 아래와 같이 TCP Flow에 대한 그래프가 나옴. 그래프에서 클라이언트와 서버측의 통신포트를 확인해보기

#31. Wireshark로 패킷 Capture하기

문서번호: 20170202-01

버전: 1.0

Date: 2017/02/02

