

1주차_환경설정 && wireshark

데이터 네트워크연구실
이현호

lee075@cs-cnu.org

Goals

- 가상환경 설치 및 구축
- Wireshark 설치 및 실행
- lua script

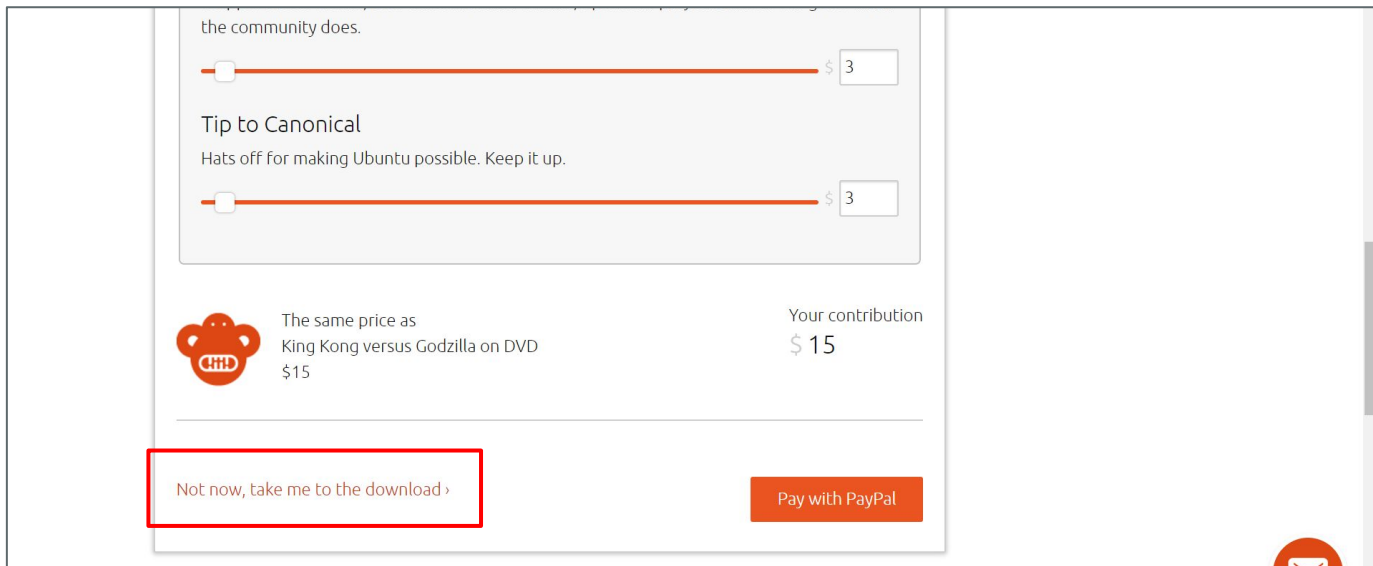
가상환경 구축

1. 실습은 전부 LINUX에서 진행
2. PC에 리눅스를 설치
3. 리눅스 가상환경을 구축
4. VM Workstation Player 12

Install


- Ubuntu 다운로드 링크 :

<https://www.ubuntu.com/download/desktop/contribute?version=16.04.3&architecture=amd64>



the community does.

Tip to Canonical
Hats off for making Ubuntu possible. Keep it up.

 The same price as
King Kong versus Godzilla on DVD
\$15

Your contribution
\$ 15

[Not now, take me to the download >](#)

[Pay with PayPal](#)

Install

- VM Player 다운로드 링크 :

https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0

VMware Cloud

PRODUCTS

SOLUTIONS

SUPPORT

DOWNLOADS

PROFESSIONAL SERVICES

PARTNER PROGRAMS

Major Version: 12.0 (latest) Minor Version: 12.5.7 (latest)

Product Downloads Drivers & Tools Open Source

Need help downloading?

VMware Workstation 12.5.7 Player for Windows 64-bit Operating Systems.

(exe | 78.28 MB)

Show Details

Download

About This Product

DESCRIPTION

VMware Workstation 12.5.7 Player

DOCUMENTATION

Release Notes

NOTES

VMware Workstation Player (FREE & PAID).

Enter a license key into the VMware Workstation Player user interface to license for commercial use and to enable the VMware Workstation Player (PAID) features.

VMware Workstation 12.5.7 Player for Linux 64-bit.

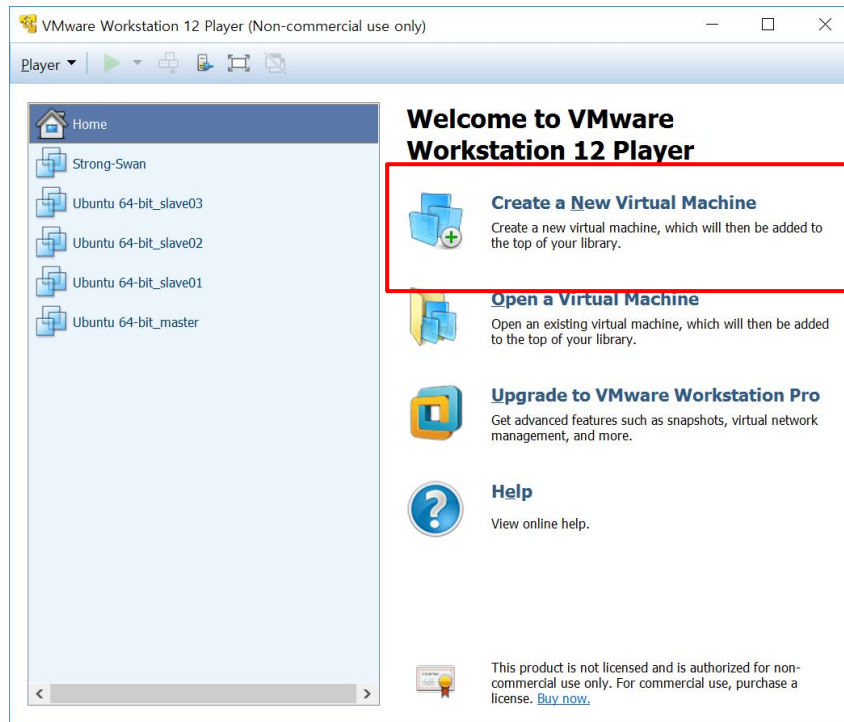
(bundle | 128.01 MB)

Show Details

Download

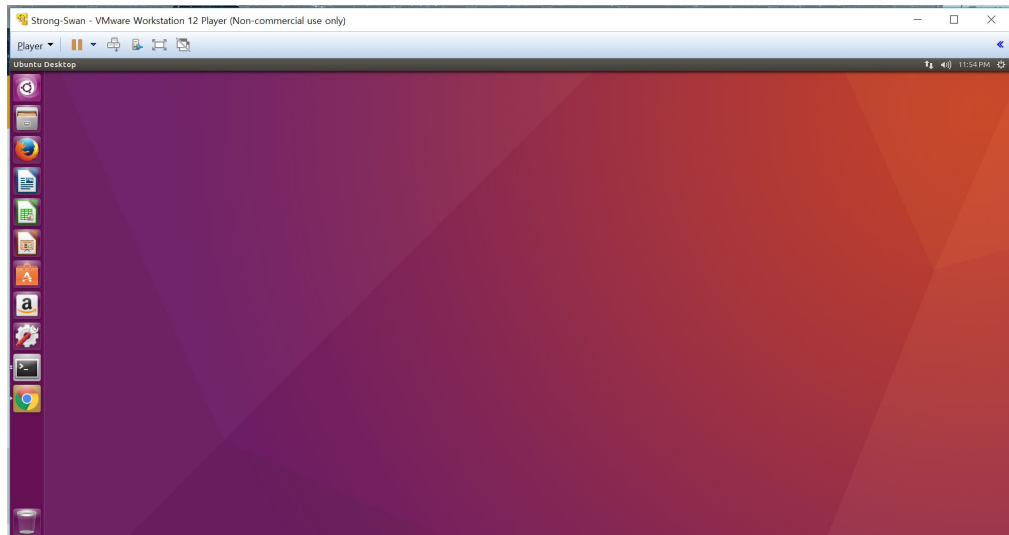
Virtual Machine

- 이미지 생성
- Ubuntu 생성
- 설치 후 설정



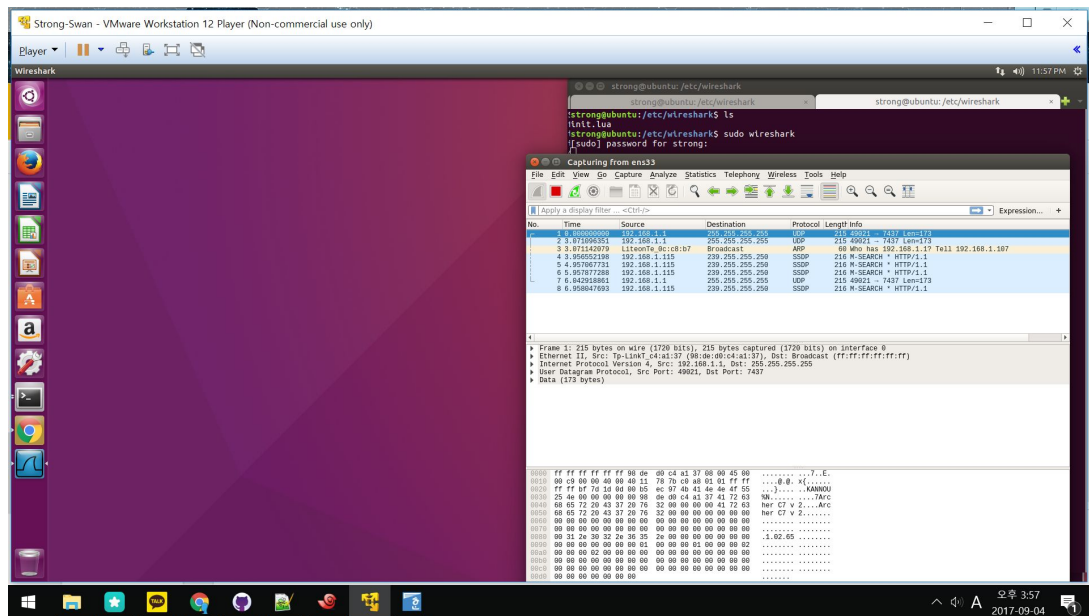
Hello Ubuntu

- 시작하면
- 터미널을 이용
- 사용하면서 적응
- 리눅스 명령어
 - http://www.mireene.com/webimg/linux_tip1.htm



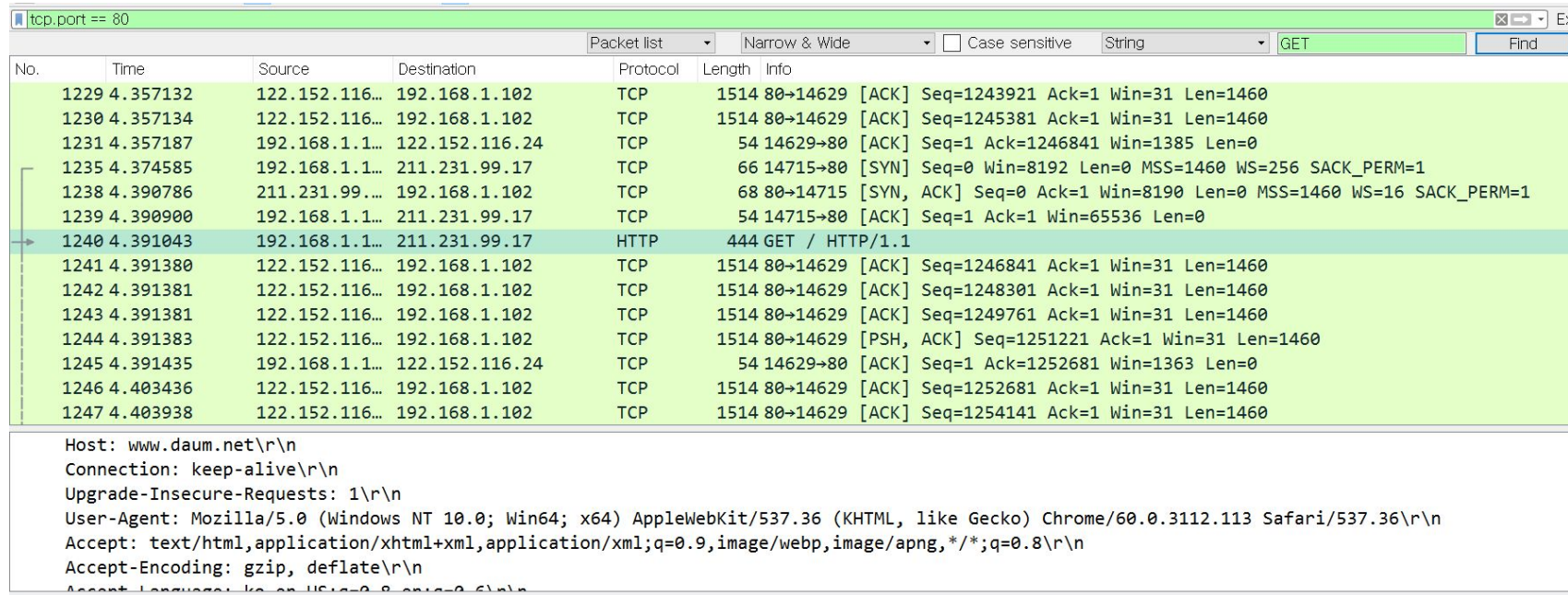
Wireshark

- 설치 : `sudo apt-get install wireshark`
- 실행 : `sudo wireshark`



How to use?

- filter를 이용(https://openmaniak.com/kr/wireshark_filters.php)
- tcp.port == 80 (http는 80포트를 이용)



tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
1229	4.357132	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1243921 Ack=1 Win=31 Len=1460
1230	4.357134	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1245381 Ack=1 Win=31 Len=1460
1231	4.357187	192.168.1.1...	122.152.116.24	TCP	54	14629→80 [ACK] Seq=1 Ack=1246841 Win=1385 Len=0
1235	4.374585	192.168.1.1...	211.231.99.17	TCP	66	14715→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1238	4.390786	211.231.99...	192.168.1.102	TCP	68	80→14715 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1460 WS=16 SACK_PERM=1
1239	4.390900	192.168.1.1...	211.231.99.17	TCP	54	14715→80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
1240	4.391043	192.168.1.1...	211.231.99.17	HTTP	444	GET / HTTP/1.1
1241	4.391380	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1246841 Ack=1 Win=31 Len=1460
1242	4.391381	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1248301 Ack=1 Win=31 Len=1460
1243	4.391381	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1249761 Ack=1 Win=31 Len=1460
1244	4.391383	122.152.116...	192.168.1.102	TCP	1514	80→14629 [PSH, ACK] Seq=1251221 Ack=1 Win=31 Len=1460
1245	4.391435	192.168.1.1...	122.152.116.24	TCP	54	14629→80 [ACK] Seq=1 Ack=1252681 Win=1363 Len=0
1246	4.403436	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1252681 Ack=1 Win=31 Len=1460
1247	4.403938	122.152.116...	192.168.1.102	TCP	1514	80→14629 [ACK] Seq=1254141 Ack=1 Win=31 Len=1460

Host: www.daum.net\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: ko-KR;q=0.9,en-US;q=0.8,en;q=0.6\r\n

What is lua script

- 루아(Lua)는 프로그래밍 언어로 가벼운 명령형/절차적 언어
- 확장 언어로 쓰일 수 있는 스크립팅 언어를 주 목적으로 설계



Scenario

- 개발을 하다보면 프로토콜을 개발
- 정상동작 하는지 혹은 데이터를 분석
- wire shark에서 보일리 없음
- 구조를 정의하고 분석

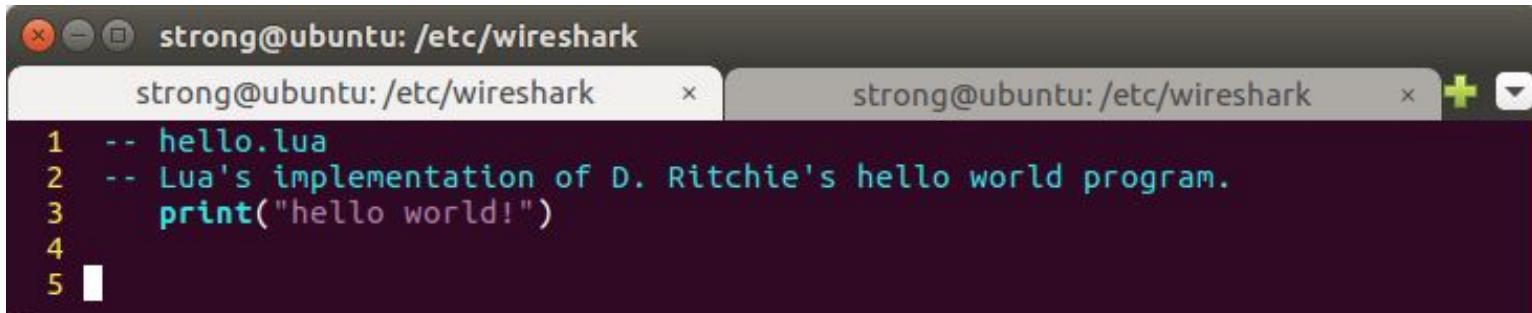
lua script

- 참고문서 : <https://wiki.wireshark.org/Lua>
- Wireshark가 스크립트를 찾을지 말지 정하게하는 두 변수 확인
 - disable_lua
 - run_user_scripts_when_superuser(Wireshark가 Super User로 실행)

```
26 -- Set disable_lua to true to disable Lua support.
27 disable_lua = true
28
29 if disable_lua then
30     return
31 end
32
33 -- If set and we are running with special privileges this setting
34 -- tells whether scripts other than this one are to be run.
35 run_user_scripts_when_superuser = true
36
37
38 -- disable potentially harmful lua functions when running superuser
39 if running_superuser then
```

hello lua script

- 잘 모르는 언어
- 그래도 HELLO WORLD는 예의
- 기초적인 스크립트 작성

A screenshot of a terminal window with a dark background. The title bar shows 'strong@ubuntu: /etc/wireshark'. There are two tabs open, both with the same title. The terminal content shows a Lua script with line numbers 1 through 5. Line 1 is a comment '-- hello.lua'. Line 2 is a comment '-- Lua's implementation of D. Ritchie's hello world program.'. Line 3 is the command 'print("hello world!")'. Line 4 is empty. Line 5 has a cursor. The text is color-coded: comments are light blue, and the print function is purple.

```
1  -- hello.lua
2  -- Lua's implementation of D. Ritchie's hello world program.
3  print("hello world!")
4
5  
```

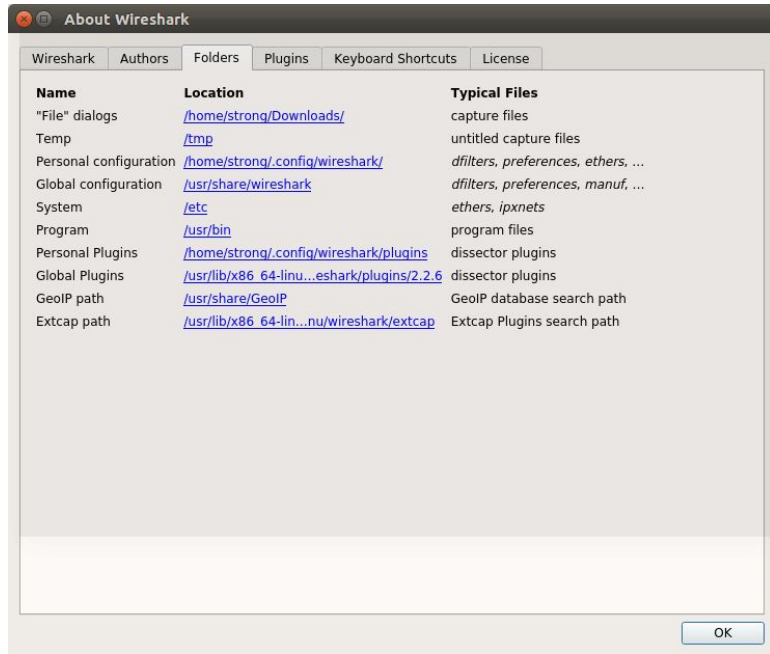
Run wire shark

- wire shark를 실행하면 !
- hello world!

```
strong@ubuntu:/etc/wireshark$ sudo wireshark  
hello world!
```

What can we do?

- 실제 lua Script를 적용해보자
- Global Plugins에 추가



structure of header

- 헤더의 구조와 코드레벨

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCode			
QDCount															
ANCount															
NSCount															
ARCount															
(Header 结构)每行 16 位。Lixin															

```
-----  
-- multiple ways to do the same thing: create a protocol field (but not register it yet)  
-- the abbreviation should always have "<myproto>." before the specific abbreviation, to avoid collisions  
local pf_transaction_id = ProtoField.new ("Transaction ID", "mydns.trans_id", ftypes.UINT16)  
local pf_flags = ProtoField.new ("Flags", "mydns.flags", ftypes.UINT16, nil, base.HEX)  
local pf_num_questions = ProtoField.uint16("mydns.num_questions", "Number of Questions")  
local pf_num_answers = ProtoField.uint16("mydns.num_answers", "Number of Answer RRs")  
local pf_num_authority_rr = ProtoField.uint16("mydns.num_authority_rr", "Number of Authority RRs")  
local pf_num_additional_rr = ProtoField.uint16("mydns.num_additional_rr", "Number of Additional RRs")
```


Compare

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.50.50	192.168.0.1	UDP	75	65282 → 65333 Len=33
2	0.006946	192.168.0.1	192.168.50.50	UDP	540	65333 → 65282 Len=498

Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
Ethernet II, Src: AmbitMic_6c:40:4e (00:d0:59:6c:40:4e), Dst: Cisco-Li_82:b2:53 (00:0c:41:82:b2:53)
Internet Protocol Version 4, Src: 192.168.50.50, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 65282, Dst Port: 65333
Data (33 bytes)
Data: 002b0100000100000000000002757304706f6f6c036e7470...
[Length: 33]

0000 00 0c 41 82 b2 53 00 d0 59 6c 40 4e 00 00 45 00 ..A..S...Y1@N..E.
0010 00 3d 0a 41 00 00 80 11 7c eb c0 a8 32 32 c0 a8 .=A....|...22..
0020 00 01 ff 02 ff 35 00 29 07 a9 00 2b 01 00 00 015.)...+....
0030 00 00 00 00 00 00 02 75 73 04 70 6f 6f 6c 03 6eu s.pool.n
0040 74 70 03 6f 72 67 00 00 01 00 01 tp.org... ..

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.50.50	192.168.0.1	MYDNS	75	Query (43) us.pool.ntp.org
2	0.006946	192.168.0.1	192.168.50.50	MYDNS	540	Response (43) us.pool.ntp.org

Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
Ethernet II, Src: AmbitMic_6c:40:4e (00:d0:59:6c:40:4e), Dst: Cisco-Li_82:b2:53 (00:0c:41:82:b2:53)
Internet Protocol Version 4, Src: 192.168.50.50, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 65282, Dst Port: 65333
MYDNS Protocol

0000 00 0c 41 82 b2 53 00 d0 59 6c 40 4e 00 00 45 00 ..A..S...Y1@N..E.
0010 00 3d 0a 41 00 00 80 11 7c eb c0 a8 32 32 c0 a8 .=A....|...22..
0020 00 01 ff 02 ff 35 00 29 07 a9 00 2b 01 00 00 015.)...+....
0030 00 00 00 00 00 00 02 75 73 04 70 6f 6f 6c 03 6eu s.pool.n
0040 74 70 03 6f 72 67 00 00 01 00 01 tp.org... ..

과제

1. Linux 실험환경 구축한 후 결과화면
2. dissector.lua 내부에 구현되어있는 프로토콜 분석 레포트

과제 제출

- 과제 제출 기한:
 - 실습 하루 전 18시
- e-learning 페이지에 제출
- 보고서 제목 : NW_학번_이름_실습번호.pdf
- 추가 첨부파일 : NW_학번_이름_실습번호.zip
 - 추가 첨부파일은 본인이 작성한 파일로 제한합니다