

네트워크 패킷 분석

BCGLAB 세미나 4/6

IT정보공학과 박수빈



LIST

01. 패킷

02. 와이어샤크

03. 패킷 분석 실습

패킷 - 패킷이란?

패킷: 데이터를 여러 개로 쪼갠 작은 조각

네트워크를 통해 전송되는 데이터는 작은 단위의 패킷으로 나뉘어 전송
수신하는 기기에서 재조립

패킷을 사용하는 이유

데이터를 패킷으로 나누지 않고 전송할 경우 대역폭을 많이 차지하여 비효율적
(두 대의 컴퓨터가 공유된 케이블을 통해 긴 비트 라인을 송/수신할 경우 이 케이블에 연결된 다른
컴퓨터는 아무 정보도 보내지 못한다.)

패킷 교환을 이용 시 여러 대의 컴퓨터가 동일한 케이블을 통해 동시에 패킷을 전송 가능

패킷 교환: 네트워킹 장비가 패킷을 독립적으로 처리하는 것.

패킷 - 패킷의 구성요소

헤더

소스 주소, 대상 주소, 프로토콜, 패킷 번호 포함
계층(OSI 7 Layer)을 지날 때마다 덧붙여짐

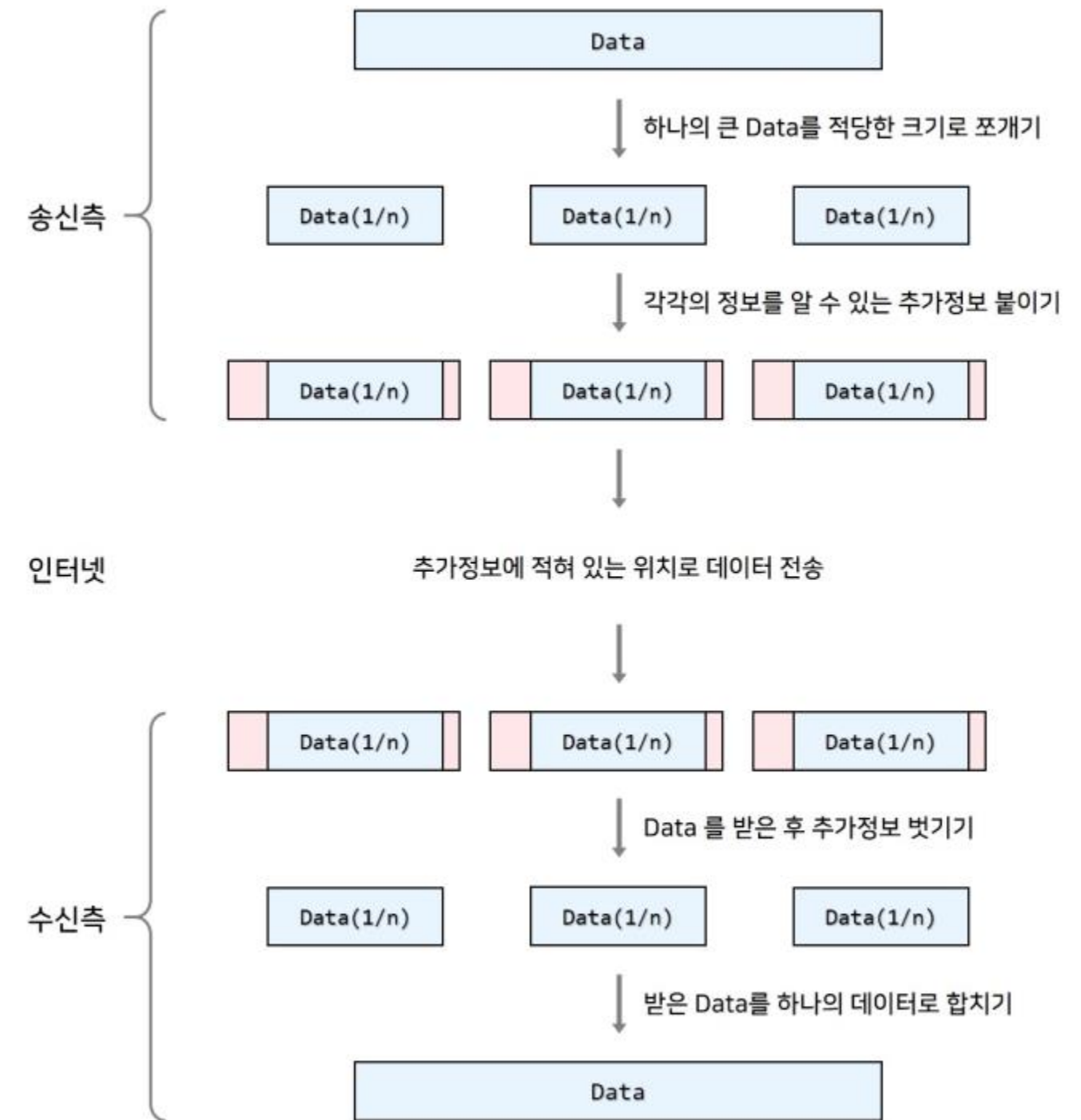
페이로드

실제 데이터

트레일러

네트워크 유형에 따라 다르다.

수신 장치에 패킷 끝까지 도달했음을 알리는 비트, 모든 패킷이 완전히 수신되었는지 확인할 수 있는 CRC(Cyclic Redundancy Check)

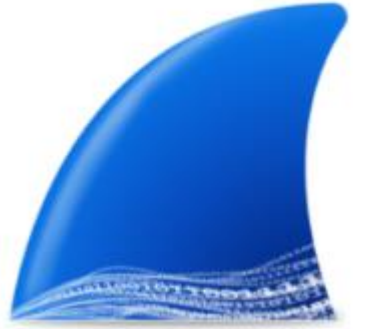


이미지 출처:

<https://velog.io/@emplam27/CS-%EA%B7%B8%EB%A6%BC%EC%9C%BC%EB%A1%9C-%EC%95%8C%EC%95%84%EB%B3%B4%EB%8A%94-%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EA%B3%84%EC%B8%B5%ED%99%94%EC%99%80-OSI-TCPIP-UDP%EC%9D%98-%ED%8A%B9%EC%A7%95%EA%B3%BC-%EC%B0%A8%EC%9D%B4%EC%A0%90>

와이어샤크

네트워크 패킷 분석 및 감시 프로그램



보안 분야 활용 예시

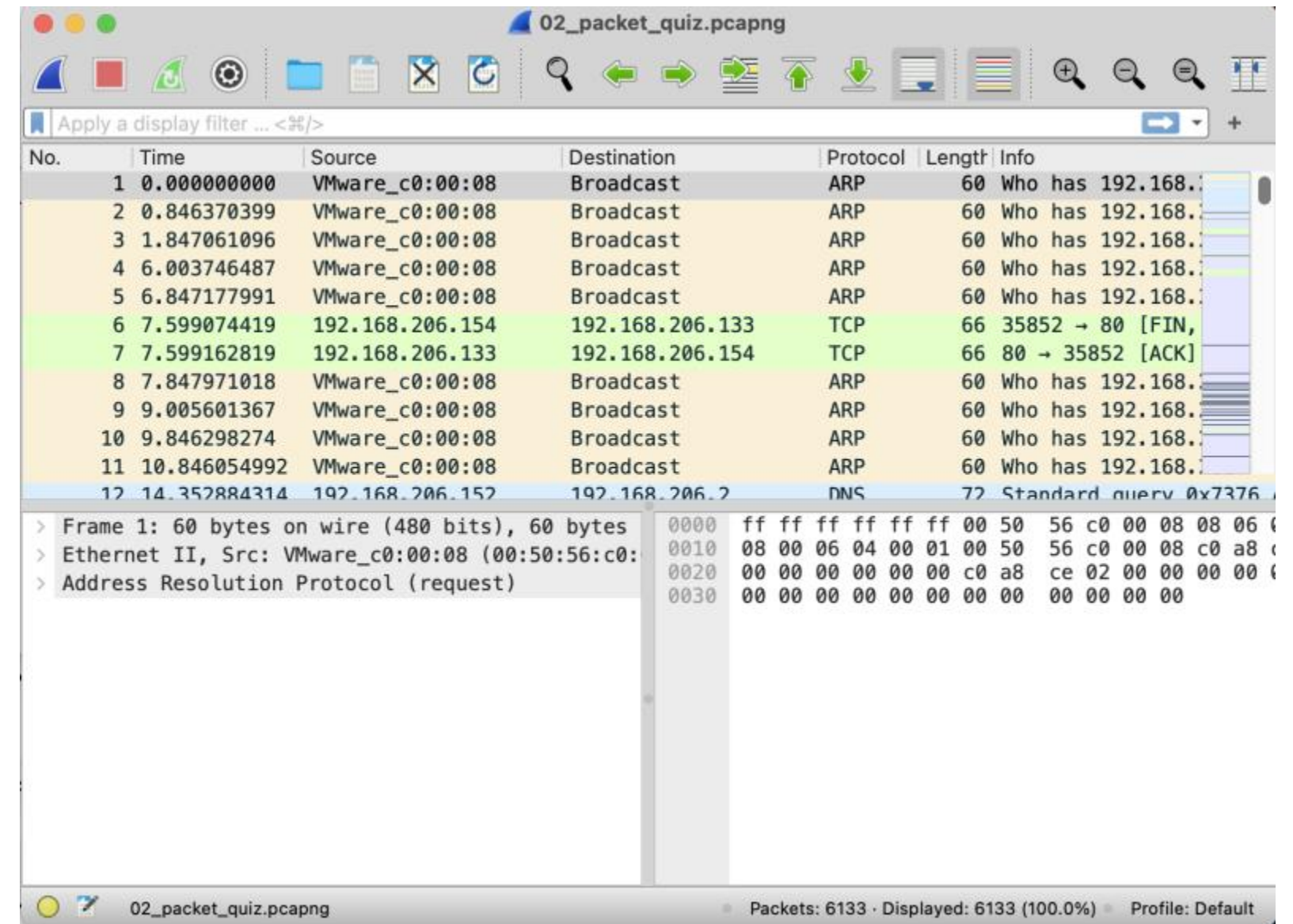
침해대응 분석

- 모니터링

모의해킹

- PC -> 서버

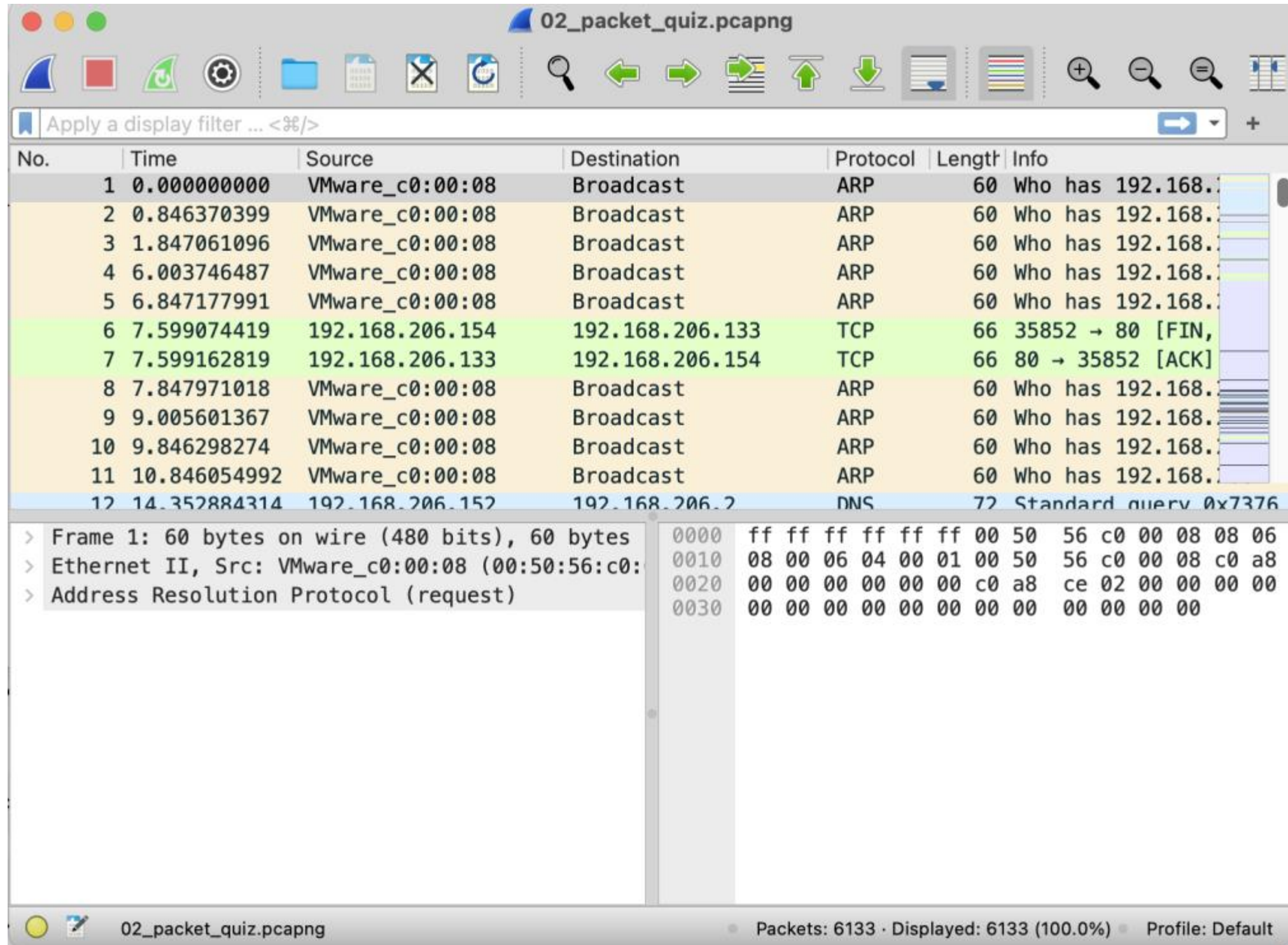
포렌식 분석



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
2	0.846370399	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
3	1.847061096	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
4	6.003746487	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
5	6.847177991	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
6	7.599074419	192.168.206.154	192.168.206.133	TCP	66	35852 -> 80 [FIN, Seq=123456789]
7	7.599162819	192.168.206.133	192.168.206.154	TCP	66	80 -> 35852 [ACK, Seq=987654321]
8	7.847971018	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
9	9.005601367	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
10	9.846298274	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
11	10.846054992	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
12	14.352884314	192.168.206.152	192.168.206.2	DNS	72	Standard query 0x7376...

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Destination: 00:00:00:00:00:00
Address Resolution Protocol (request)

와이어샤크



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
2	0.846370399	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
3	1.847061096	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
4	6.003746487	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
5	6.847177991	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
6	7.599074419	192.168.206.154	192.168.206.133	TCP	66	35852 → 80 [FIN, Seq=123456789]
7	7.599162819	192.168.206.133	192.168.206.154	TCP	66	80 → 35852 [ACK, Seq=987654321]
8	7.847971018	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
9	9.005601367	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
10	9.846298274	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
11	10.846054992	VMware_c0:00:08	Broadcast	ARP	60	Who has 192.168.1.1
12	14.352884314	192.168.206.152	192.168.206.2	DNS	72	Standard query 0x7376

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Address Resolution Protocol (request) [ethertype 0800]

←

NO: 수집된 패킷의 순서
Time: 시간대
Source: 출발지 주소
Destination: 도착지 주소
Protocol: 프로토콜 type
Length: 패킷 길이
Info: 패킷 정보

패킷 분석 실습

1. 공격에 성공한 공격자 IP는?
2. 공격자 이외 서비스에 접근한 IP는?
3. 공격자가 시스템에 침투하기 위해 접근한 서비스 포트는?
4. 공격자가 어떤 취약점을 이용한 것인지 서술하시오.
5. 공격자가 올린 웹shell의 이름과 md5 해시 값은?
6. 공격자가 웹shell을 올린 뒤에 사용한 명령어를 찾는 대로 작성하시오.

패킷 분석 실습

1. 공격에 성공한 공격자 IP는? / 2. 공격자 이외 서비스에 접근한 IP는?

Ethernet · 14 IPv4 · 41 IPv6 · 1 TCP · 78 UDP · 124									
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Bytes B → A
192.168.206.152	50054	192.168.206.133	80	73	35.755 KiB	9	30	3.945 KiB	1.016 KiB
192.168.206.152	50056	192.168.206.133	80	36	30.638 KiB	10	18	3.042 KiB	1.016 KiB
192.168.206.152	44982	192.168.206.133	8180	180	162.781 KiB	12	83	40.631 KiB	1.016 KiB
192.168.206.152	44984	192.168.206.133	8180	22	15.355 KiB	13	11	1.399 KiB	1.016 KiB
192.168.206.152	44986	192.168.206.133	8180	20	6.545 KiB	14	10	1.332 KiB	1.016 KiB
192.168.206.152	44992	192.168.206.133	8180	11	3.351 KiB	37	6	1,016 bytes	1.016 KiB
192.168.206.152	44994	192.168.206.133	8180	18	18.117 KiB	40	10	1.640 KiB	1.016 KiB
192.168.206.154	35852	192.168.206.133	80	2	132 bytes	0	1	66 bytes	1.016 KiB
192.168.206.154	35853	192.168.206.133	80	113	151.653 KiB	18	58	6.468 KiB	1.016 KiB
192.168.206.154	35854	192.168.206.133	80	58	68.477 KiB	19	32	3.949 KiB	1.016 KiB
192.168.206.154	35855	192.168.206.133	80	51	73.652 KiB	20	32	3.927 KiB	1.016 KiB
192.168.206.154	35856	192.168.206.133	80	48	46.739 KiB	21	26	3.430 KiB	1.016 KiB
192.168.206.154	35857	192.168.206.133	80	24	12.464 KiB	22	16	2.870 KiB	1.016 KiB
192.168.206.154	35858	192.168.206.133	80	54	80.451 KiB	23	36	3.737 KiB	1.016 KiB
192.168.206.154	35859	192.168.206.133	80	15	3.411 KiB	27	8	1.394 KiB	1.016 KiB
192.168.206.154	35861	192.168.206.133	80	246	466.333 KiB	29	96	7.979 KiB	1.016 KiB
192.168.206.154	35867	192.168.206.133	80	17	4.039 KiB	35	10	2.029 KiB	1.016 KiB
192.168.206.154	35868	192.168.206.133	80	14	2.814 KiB	36	8	1.403 KiB	1.016 KiB
192.168.206.133	34697	192.168.206.152	8989	10	698 bytes	38	6	423 bytes	1.016 KiB

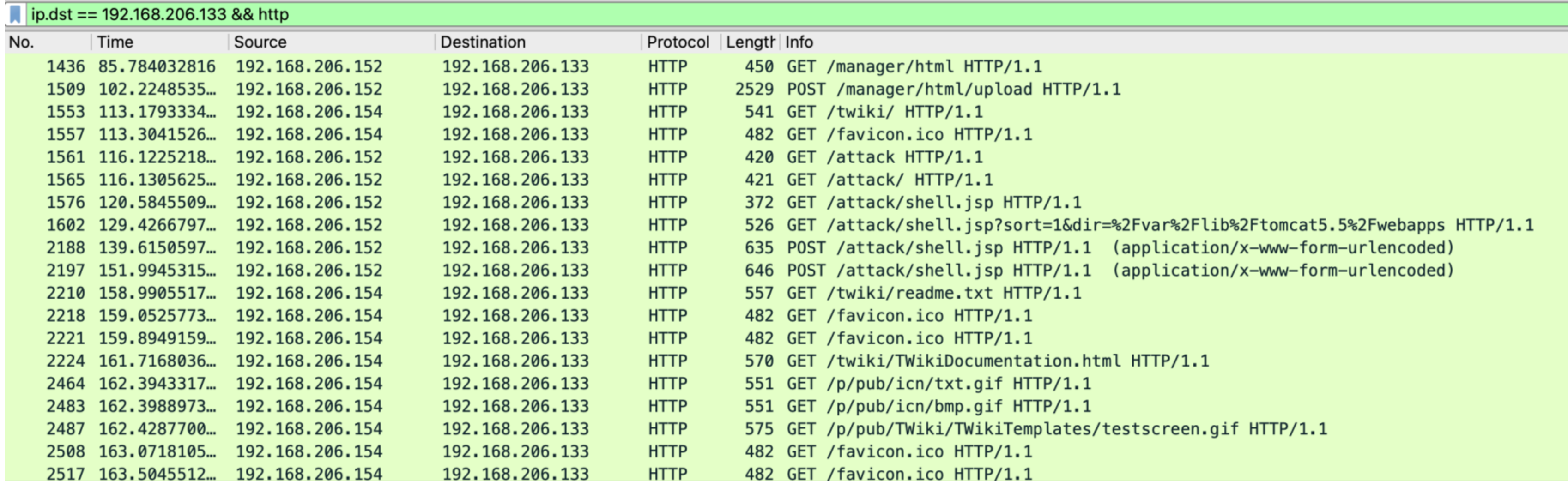
192.168.206.152 -> 80

192.168.206.152 -> 8180

192.168.206.154 -> 80

패킷 분석 실습

1. 공격에 성공한 공격자 IP는? / 2. 공격자 이외 서비스에 접근한 IP는?



ip.dst == 192.168.206.133 && http						
No.	Time	Source	Destination	Protocol	Length	Info
1436	85.784032816	192.168.206.152	192.168.206.133	HTTP	450	GET /manager/html HTTP/1.1
1509	102.2248535...	192.168.206.152	192.168.206.133	HTTP	2529	POST /manager/html/upload HTTP/1.1
1553	113.1793334...	192.168.206.154	192.168.206.133	HTTP	541	GET /twiki/ HTTP/1.1
1557	113.3041526...	192.168.206.154	192.168.206.133	HTTP	482	GET /favicon.ico HTTP/1.1
1561	116.1225218...	192.168.206.152	192.168.206.133	HTTP	420	GET /attack HTTP/1.1
1565	116.1305625...	192.168.206.152	192.168.206.133	HTTP	421	GET /attack/ HTTP/1.1
1576	120.5845509...	192.168.206.152	192.168.206.133	HTTP	372	GET /attack/shell.jsp HTTP/1.1
1602	129.4266797...	192.168.206.152	192.168.206.133	HTTP	526	GET /attack/shell.jsp?sort=1&dir=%2Fvar%2Flib%2Ftomcat5.5%2Fwebapps HTTP/1.1
2188	139.6150597...	192.168.206.152	192.168.206.133	HTTP	635	POST /attack/shell.jsp HTTP/1.1 (application/x-www-form-urlencoded)
2197	151.9945315...	192.168.206.152	192.168.206.133	HTTP	646	POST /attack/shell.jsp HTTP/1.1 (application/x-www-form-urlencoded)
2210	158.9905517...	192.168.206.154	192.168.206.133	HTTP	557	GET /twiki/readme.txt HTTP/1.1
2218	159.0525773...	192.168.206.154	192.168.206.133	HTTP	482	GET /favicon.ico HTTP/1.1
2221	159.8949159...	192.168.206.154	192.168.206.133	HTTP	482	GET /favicon.ico HTTP/1.1
2224	161.7168036...	192.168.206.154	192.168.206.133	HTTP	570	GET /twiki/TWikiDocumentation.html HTTP/1.1
2464	162.3943317...	192.168.206.154	192.168.206.133	HTTP	551	GET /p/pub/icn/txt.gif HTTP/1.1
2483	162.3988973...	192.168.206.154	192.168.206.133	HTTP	551	GET /p/pub/icn/bmp.gif HTTP/1.1
2487	162.4287700...	192.168.206.154	192.168.206.133	HTTP	575	GET /p/pub/TWiki/TWikiTemplates/testscreen.gif HTTP/1.1
2508	163.0718105...	192.168.206.154	192.168.206.133	HTTP	482	GET /favicon.ico HTTP/1.1
2517	163.5045512...	192.168.206.154	192.168.206.133	HTTP	482	GET /favicon.ico HTTP/1.1

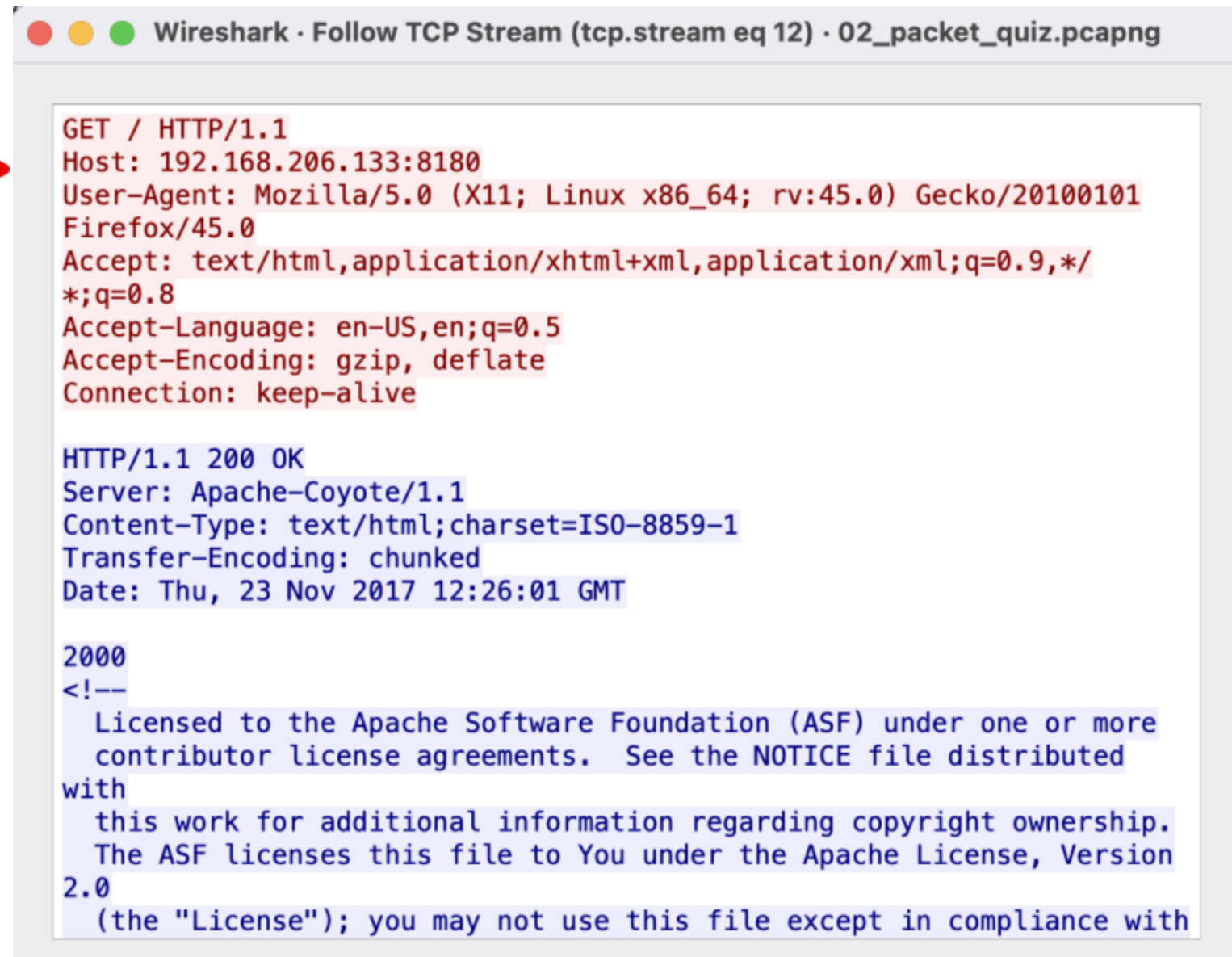
ip.dst == 192.168.206.133 && http 조건문

info에서 attack/shell.jsp 등 확인 가능

1. 192.168.206.152

2. 192.168.206.154

패킷 분석 실습 3. 공격자가 시스템에 침투하기 위해 접근한 서비스 포트는?

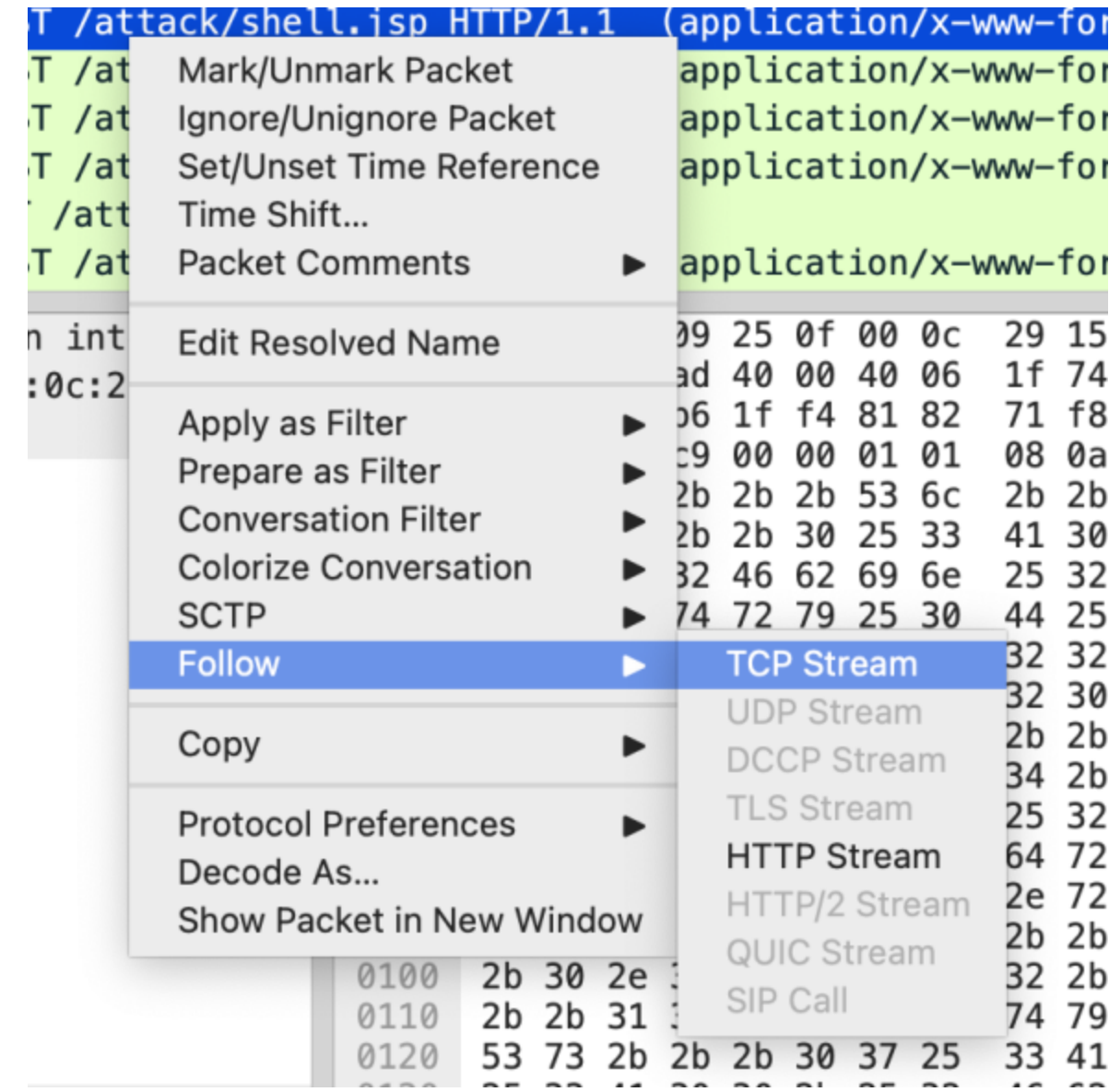


Wireshark · Follow TCP Stream (tcp.stream eq 12) · 02_packet_quiz.pcapng

```
GET / HTTP/1.1
Host: 192.168.206.133:8180
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Thu, 23 Nov 2017 12:26:01 GMT

2000
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed
with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version
2.0
(the "License"); you may not use this file except in compliance with
```



```
T /attack/shell.jsp HTTP/1.1 (application/x-www-form-urlencoded)
T /at Mark/Unmark Packet application/x-www-form-urlencoded
T /at Ignore/Unignore Packet application/x-www-form-urlencoded
T /at Set/Unset Time Reference application/x-www-form-urlencoded
/att Time Shift...
T /at Packet Comments application/x-www-form-urlencoded

n int Edit Resolved Name 09 25 0f 00 0c 29 15
:0c:2 ad 40 00 40 06 1f 74
06 1f f4 81 82 71 f8
c9 00 00 01 01 08 0a
2b 2b 2b 53 6c 2b 2b
2b 2b 30 25 33 41 30
32 46 62 69 6e 25 32
74 72 79 25 30 44 25
32 32
32 30
2b 2b
34 2b
25 32
64 72
2e 72
2b 2b
32 2b
74 79
25 33 41
```

Follow - TCP Stream

포트번호 8180

패킷 분석 실습 4. 공격자가 어떤 취약점을 이용한 것인지 서술하시오.

GET / HTTP/1.1 Host: 192.168.206.133:8180 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Connection: keep-alive HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Content-Type: text/html; charset=ISO-8859-1 Transfer-Encoding: chunked Date: Thu, 23 Nov 2017 12:26:01 GMT 2000



Apache Tomcat/5.5



The Apache Software Foundation

Administration

Status
Tomcat Administration
Tomcat Manager

Documentation

Release Notes
Change Log
Tomcat Documentation

Tomcat Online

Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC

Examples

JSP Examples
Servlet Examples
WebDAV capabilities

Miscellaneous

Sun's Java Server Pages Site
Sun's Servlet Site

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

`$CATALINA_HOME/webapps/ROOT/index.jsp`

where "\$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

- users@tomcat.apache.org for general questions related to configuring 1f4 and using Tomcat
- dev@tomcat.apache.org for developers working on Tomcat

Thanks for using Tomcat!



Copyright © 1999-2005 Apache Software Foundation
All Rights Reserved

0 GET /tomcat.gif HTTP/1.1 Host: 192.168.206.133:8180 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0 Accept: image/png,image/*;q=0.8,*/*;q=0.5 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Referer: http://192.168.206.133:8180/ Connection: keep-alive HTTP/1.1 200 OK Server: Apache-Coyote/1.1 ETag: W/"1934-1228677438000" Last-Modified: Sun, 07 Dec 2008 19:17:18 GMT Content-Type: image/gif Content-Length: 1934 Date: Thu, 23 Nov 2017 12:26:01 GMT GIF89a...#I...W..p.q&.....GB4.. ..ok]..b..u.....\.....I..8..-.)\ (Z.a.h.r..*.*?.;8.....s...JG.IM..C..*ID..-w.18...K.....L.X.... t.; u...;~.8...-1.0...f...s...5 ..t.6.....xD....9!..lI...35l..ydfN.CL...3..k ...7..%.M.].I.X/...<|.baN@.....dT\...8....DK.f.e....^....B ..TT.I.. ..QjX.....{C.....JpM.....}..... t.Q. .e.V...@..O.D..zB..b...Y...9.b..4..-.....NiC\$.....o9\$..._H'.Z.....o7.....Q.a..(b).....K....^S..D#T.' U/.0.....}%=.....<=0.5.}.....R .. p.h#4..\$. \$@X@B...@.1.4....#..K.z@g..T:D..Un..4+&.Q.1.S.:9..EI....._P%.Ce...

TCP Stream을 HTML 파일로 저장 (test.html)

Tomcat 관리자 페이지에 들어가서 웹셀 업로드

패킷 분석 실습 5. 공격자가 올린 웹shell의 이름과 md5 해시 값은?

Packet	Hostname	Content Type	Size	Filename
1509	192.168.206.133:8180	multipart/form-data	15 kB	upload
▼ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----82102593518880132671547018732\r\n"				
[Type: multipart/form-data]				
First boundary: -----82102593518880132671547018732\r\n				
▼ Encapsulated multipart part: (application/octet-stream)				
Content-Disposition: form-data; name="deployWar"; filename="attack.war"\r\n				
Content-Type: application/octet-stream\r\n\r\n				
▼ Data (15007 bytes)				
Data: 504b03041400080008007755b644000000000000000000000000090004004d4554412d49...				
[Length: 15007]				

< > **attack**



META-INF



shell.jsp

```
[humata@humataui-MacBookPro ~ % md/Users/humata/Documents/까치/bcg/attack/shell.jsp
MD5 (/Users/humata/Documents/까치/bcg/attack/shell.jsp) = 189f187c34c1d0eba2ab3562f362be41
```

data를 attack.war 파일로 저장

터미널에서 md5 해시 값 획득

패킷 분석 실습 6. 공격자가 웹쉘을 올린 뒤에 사용한 명령어를 찾는 대로 작성하시오.

192.168.206.133:8180 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0 Accept: */* Accept-Language: en-US Accept-Encoding: gzip, deflate Connection: keep-alive Host: 192.168.206.133:8180/attack/shell.jsp Cookie: JSESSIONID=BC517A3BBE51862B21E7125C74AC13DE Connection: keep-alive From: tomcat5.5%2Fwebapps%2Fattack&command=ps+-aux&Submit=Launch&sort=1 HTTP/1.1 200 OK Server: Apache/2.4.6-ubuntu

Perhaps a bogus '-'? See <http://procps.sf.net/faq.html>

test.html

ps aux: 프로세스 정보를 알려주는 명령어

THANK YOU