



네트워크 패킷에서 이미지 복원하기



WireShark 실습





목적 및 개요

- ❑ Wireshark 캡처 파일에서 이미지 추출 절차 소개
- ❑ JPEG 시그니처 기반 수동 복원 실습 포함
- ❑ 디지털 포렌식, 분석 실무에 필요한 기본기 학습 목적





실습 파일 준비

□ 샘플 캡처 파일 다운로드

- <https://wiki.wireshark.org/samplecaptures>
- <https://creamerburger.tistory.com/72>
 - 다운받고 압축풀기 귀찮으신 분들 제 티스토리에서 받아주세요
- 예시: http_witp_jpegs.cap

□ Wireshark에서 파일 열기





실습 파일 준비

□ 파일 > 열기

http_wftp_pgegs.cap

파일(F) 편집(E) 보기(V) 위젯(W) 캡처(C) 분석(A) 통계(S) 콘솔(O) 우선(U) 도구(T) 도움말(H)

표시 필터 적용 <<ON>>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.101	10.1.1.1	TCP	42	3177 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM
2	0.000051	10.1.1.1	10.1.1.101	TCP	42	80 → 3177 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
3	0.000097	10.1.1.101	10.1.1.1	TCP	54	3177 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.013660	10.1.1.101	10.1.1.1			
5	0.014730	10.1.1.1	10.1.1.101			
6	0.052280	10.1.1.1	10.1.1.101			
7	0.052346	10.1.1.1	10.1.1.101			
8	0.052407	10.1.1.101	10.1.1.1			
9	0.121705	10.1.1.101	209.225.11.237			
10	0.126302	10.1.1.101	10.1.1.1			
11	0.130700	10.1.1.1	10.1.1.101			
12	0.164255	209.225.11.237	10.1.1.101			
13	0.164322	10.1.1.101	209.225.11.237			
14	0.164820	10.1.1.101	209.225.11.237			
15	0.164962	209.225.11.237	10.1.1.101			
16	0.164949	10.1.1.101	209.225.11.237			
17	0.924215	209.225.11.237	10.1.1.101			
18	0.953850	209.225.11.237	10.1.1.101			
19	0.956640	209.225.11.237	10.1.1.101			
20	0.956770	10.1.1.101	209.225.11.237			
21	0.959512	209.225.11.237	10.1.1.101			
22	0.970120	10.1.1.101	209.225.11.237			
23	1.108834	10.1.1.101	209.225.11.237			
24	1.109417	10.1.1.101	209.225.0.6			

Frame 1: 62 bytes on wire (496 bits), 62 bytes captured on interface 0
Ethernet II, Src: SPKNetworks_22:5a:03 (00:04:e2:22:5a:03), Dst: KYE_28:10:cdf (00:c8:ef:28:10:cdf)
Source: SPKNetworks_22:5a:03 (00:04:e2:22:5a:03)
Type: IPv4 (0x0800)
[Stream Index: 0]
Internet Protocol Version 4, Src: 10.1.1.101, Dst: 209.225.11.237
Transmission Control Protocol, Src Port: 3177, Dst Port: 80

Wireshark - 캡처 파일 열기

C:\Users\User\Downloads\http_wftp_pgegs.cap

이름	크기	형식	수정된 날짜
http_wftp_pgegs.cap	319 KB	내_거	2024-1-20 7:40

파일 이름: http_wftp_pgegs.cap 열기(O) 취소

파일 형식: 모든 캡처 파일 도움말

파일 형식을 자동으로 감지: 형식: Wireshark/tcpdump! - pcap
크기: 319 KB, 마지막 바이트 483개
시작/종료: 2004-11-20 07:29:14 / 00:00:11

열기 필터: 열기 필터 적용





HTTP 패킷 필터링

- 필터: http 입력
- 이미지 포함 HTTP 응답 확인
- 크기 큰 패킷, 응답 본문에 이미
지 포함 가능성 높음
- 하단에 이미지 관련된 패킷 존재

No.	Time	Source	Destination	Protocol	Length	Info
4	0.813669	10.1.1.101	10.1.1.1	HTTP	536	GET / HTTP/1.1
6	0.832289	10.1.1.1	10.1.1.101	HTTP	489	HTTP/1.1 200 OK (text/html)
16	0.849949	10.1.1.101	209.225.11.237	HTTP	487	POST /scripts/cws/voms.asp HTTP/1.1 (application/x-www-form-urlencoded) [Continuation]
17	0.850450	209.225.11.237	10.1.1.101	HTTP	536	200 OK (text/html) [Continuation]
31	1.263955	10.1.1.101	10.1.1.1	HTTP	428	GET /vomsidn/index.html HTTP/1.1
38	1.292367	10.1.1.1	10.1.1.101	HTTP	279	HTTP/1.1 200 OK (text/html)
48	1.403685	10.1.1.101	10.1.1.1	HTTP	851	GET /vomsidn/images/bg2.jpg HTTP/1.1
50	1.404938	10.1.1.101	10.1.1.1	HTTP	854	GET /vomsidn/images/sydney.jpg HTTP/1.1
61	1.416168	10.1.1.1	10.1.1.101	HTTP	1320	HTTP/1.1 200 OK (JPEG 3F2F image)
72	1.424538	10.1.1.1	10.1.1.101	HTTP	824	HTTP/1.1 200 OK (JPEG 3F2F image)
82	1.581785	10.1.1.101	209.225.0.6	HTTP	1211	GET /site=126885/vnum=opera1/bin=1/oid=10090285
84	1.582133	10.1.1.101	209.225.0.6	HTTP	1211	GET /site=126885/vnum=opera2/bin=1/oid=10090867
86	1.582407	10.1.1.101	209.225.0.6	HTTP	1211	GET /site=126885/vnum=opera3/bin=1/oid=10092112
90	1.589370	10.1.1.101	209.225.0.6	HTTP	1211	GET /site=126885/vnum=opera4/bin=1/oid=10093085
100	1.594517	209.225.0.6	10.1.1.101	HTTP	1301	[TCP Previous segment not captured] Continuation
101	2.189739	209.225.0.6	10.1.1.101	HTTP	1301	[TCP Previous segment not captured] Continuation
110	2.222221	209.225.0.6	10.1.1.101	HTTP	1301	[TCP Previous segment not captured] Continuation
128	2.509526	10.1.1.101	209.225.0.6	HTTP	1267	GET /site=0000127789/vnum=0000162763/genr=1/logs=
133	2.504153	10.1.1.101	209.225.0.6	HTTP	1267	GET /site=0000127789/vnum=0000162763/genr=1/logs=
137	2.862516	209.225.0.6	10.1.1.101	HTTP	1301	[TCP Previous segment not captured] Continuation
145	2.845244	10.1.1.101	209.225.0.6	HTTP	1267	GET /site=0000127789/vnum=0000162766/genr=1/logs=
157	3.263972	10.1.1.101	10.1.1.1	HTTP	855	GET /vomsidn/dagbok/dagbok.html HTTP/1.1
159	3.266180	10.1.1.1	10.1.1.101	HTTP	746	HTTP/1.1 200 OK (text/html)
189	3.399757	10.1.1.101	209.225.0.6	HTTP	1267	GET /site=0000127789/vnum=0000162763/genr=1/logs=



HTTP 스트림 저장

□ 대상 패킷 우클릭 → 따라가기 → HTTP 스트림

The screenshot shows the Wireshark interface with a packet capture of an HTTP stream. The packet list pane shows a packet of type HTTP with status 200 OK. The packet details pane shows the Hypertext Transfer Protocol section. The packet bytes pane shows the raw data. A context menu is open over the packet, with 'Follow in HTTP Stream' selected.

No.	Time	Source	Destination	Protocol	Length	Info
40	1.379484	10.1.1.101	10.1.1.1	TCP	62	3189 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM
41	1.379927	10.1.1.1	10.1.1.101	TCP	62	80 → 3189 [SYN, ACK] Seq=0 Ack=1 Min=5840 Len=0 MSS=1460 SACK_PERM
42	1.379970	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
48	1.403683	10.1.1.101	10.1.1.1	HTTP	651	GET /Webadmin/images/bg2.jpg HTTP/1.1
49	1.404885	10.1.1.1	10.1.1.101	TCP	60	80 → 3189 [ACK] Seq=1 Ack=598 Min=6567 Len=0
52	1.408298	10.1.1.1	10.1.1.101	TCP	1514	80 → 3189 [ACK] Seq=1 Ack=598 Min=6567 Len=1460 [TCP PDU reassembled in 61]
53	1.409540	10.1.1.1	10.1.1.101	TCP	1514	80 → 3189 [ACK] Seq=1461 Ack=598 Min=6567 Len=1460 [TCP PDU reassembled in 61]
54	1.409607	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [ACK] Seq=598 Ack=2921 Min=65535 Len=0
56	1.412710	10.1.1.1	10.1.1.101	TCP	1514	80 → 3189 [ACK] Seq=2921 Ack=598 Min=6567 Len=1460 [TCP PDU reassembled in 61]
57	1.412812	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [ACK] Seq=598 Ack=4381 Min=65535 Len=0
58	1.413969	10.1.1.1	10.1.1.101	TCP	1514	80 → 3189 [ACK] Seq=4381 Ack=598 Min=6567 Len=1460 [TCP PDU reassembled in 61]
59	1.415274	10.1.1.1	10.1.1.101	TCP	1514	80 → 3189 [ACK] Seq=5841 Ack=598 Min=6567 Len=1460 [TCP PDU reassembled in 61]
60	1.415534	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [ACK] Seq=598 Ack=7301 Min=65535 Len=0
61	1.416360	10.1.1.1	10.1.1.101	HTTP	1320	HTTP/1.1 200 OK (JPG)
62	1.416444	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [ACK] Seq=...
93	1.543370	10.1.1.101	10.1.1.1	TCP	54	3189 → 80 [FIN, ACK]

Frame 61: 1320 bytes on wire (10560 bits), 1320 bytes captured (10560 bits) on Ethernet II, Src: KYE_20:0c:cd (00:c0:df:20:0c:cd), Dst: SHCNetworks_22:5a:03 (00:04:e2:22:5a:03)

Internet Protocol Version 4, Src: 10.1.1.1, Dst: 10.1.1.101

Transmission Control Protocol, Src Port: 80, Dst Port: 3189, Seq: 7301, Ack: 598, Len: 0

6 Reassembled TCP Segments (8566 bytes): #52(1460), #53(1460), #56(1460), #58(1460), #59(1460), #60(1460)

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\nDate: Sat, 20 Nov 2004 10:21:07 GMT\r\nServer: Apache/2.0.40 (Red Hat Linux)\r\nLast-Modified: Fri, 12 Jan 2001 05:00:00 GMT\r\nETag: "06a4f-2059-5a467400"\r\nAccept-Ranges: bytes\r\nContent-Length: 8281\r\nConnection: close\r\nContent-Type: image/jpeg\r\nX-Padi: avoid browser bug\r\n\r\n

Context menu options:

- 선택 항목 마크/해제(M) Ctrl+M
- 선택 항목 무시/재제(O) Ctrl+D
- 시간 읽음 설정/해제 Ctrl+T
- 파일시프트... Ctrl+Shift+T
- 패킷 추적
- 해석된 이름 편집
- 필터로 적용
- 필터로 언어
- 대화 필터
- 대화 색상화
- SCTP
- 따라가기 Ctrl+Alt+Shift+H
- 복사 Ctrl+Alt+Shift+T
- 프로토콜 설정
- 다른 형식으로 디코딩...
- 새 창에 패킷 표시(W)





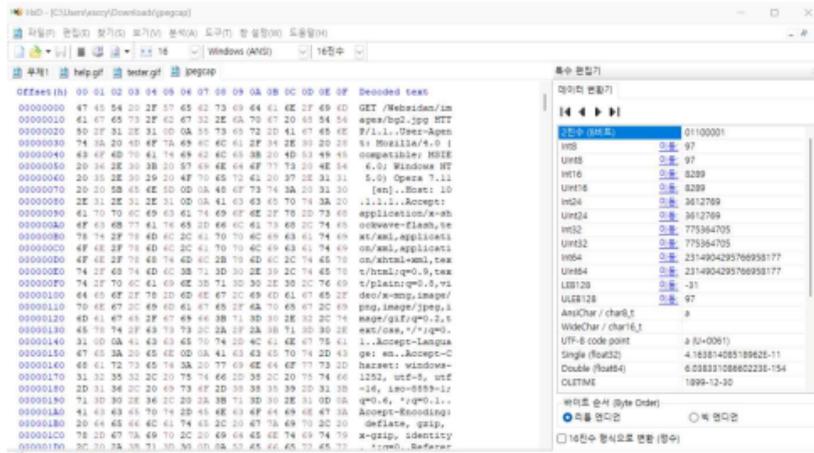
Hex Editor 설치 및 사용

❑ HxD 다운로드

- <https://mh-nexus.de/en/downloads.php?product=HxD20>

❑ 파일 열기 또는 새 창 생성

❑ Wireshark에서 저장한 Raw 파일 열기





파일 시그니처 개념과 구조

□ 파일 시그니처(Signature)

- 특정 파일 포맷이 가지는 고유한 바이트 패턴
- 일반적으로 파일 시작(Header) 또는 파일 끝(Footer)에 존재
- Hex 편집기나 포렌식 도구에서 파일 식별 및 복원 근거로 사용

□ 구조 분류

- 고정형 시그니처: 위치와 바이트값이 명확히 정의됨 (예: JPEG, PNG)
- 가변형 시그니처: 일부 포맷은 끝 시그니처가 없음, 또는 CRC 등으로 결정됨 (예: ZIP, PNG)





파일 시그니처 개념과 구조

□ Hex에서의 표현 방식 예시

- Hex: FF D8 FF → JPEG 시작
- ASCII: %PDF → PDF 시작

□ 시그니처 사용 목적

- 데이터 손상 복구
- 메모리/디스크 덤프에서 유효 파일 추출
- 악성코드 은닉 탐지 (확장자 위장이 아닌 시그니처 확인)





JPEG 포맷 분석 및 시그니처 탐색

- JPEG 파일 구조
 - JPEG은 세그먼트 기반 포맷으로, 각 마커(marker)는 0xFF로 시작
 - 대표적인 마커:
 - FF D8 → Start Of Image (SOI)
 - FF D9 → End Of Image (EOI)
- 파일 시작 시그니처
 - FF D8 FF
 - FF D8 → SOI
 - 뒤따르는 FF는 JFIF, EXIF 등 메타 정보 시작 마커 (FF E0, FF E1 등)
- 파일 종료 시그니처
 - FF D9
 - 모든 JPEG은 이 바이트로 종료
 - 이후 데이터는 무시하거나 삭제 가능





기타 포맷 시그니처 비교

포맷	시작 시그니처 (Header)	종료 시그니처 (Footer)	설명
JPEG	FF D8 FF	FF D9	가장 흔한 이미지 포맷
PNG	89 50 4E 47 0D 0A 1A 0 A	CRC 끝	비손실 압축, 시그니처 고유
GIF87a	47 49 46 38 37 61	00 3B	구형 GIF
GIF89a	47 49 46 38 39 61	00 3B	애니메이션 지원
PDF	25 50 44 46	25 45 4F 46	텍스트 기반 문서 포맷
ZIP	50 4B 03 04	50 4B 05 06	압축 포맷, DOCX 계열 기반
EXE	4D 5A (MZ)	없음	윈도우 실행 파일

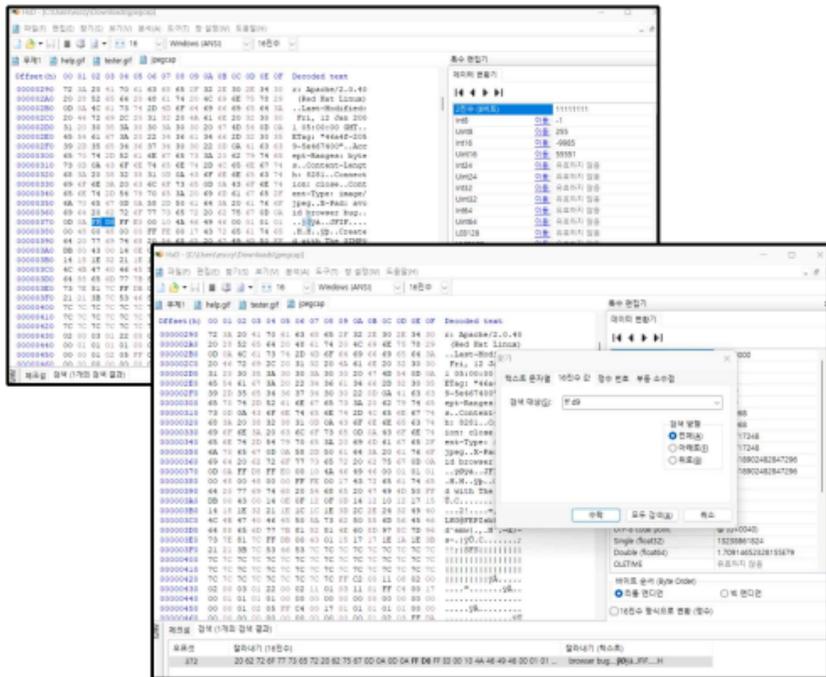




JPEG 시그니처 분석

❑ JPEG 시작 시그니처 (Header):
FF D8 FF (Ctrl + F 버튼으로 검색 후, 16진수 탭 선택 > FF D8 검색)

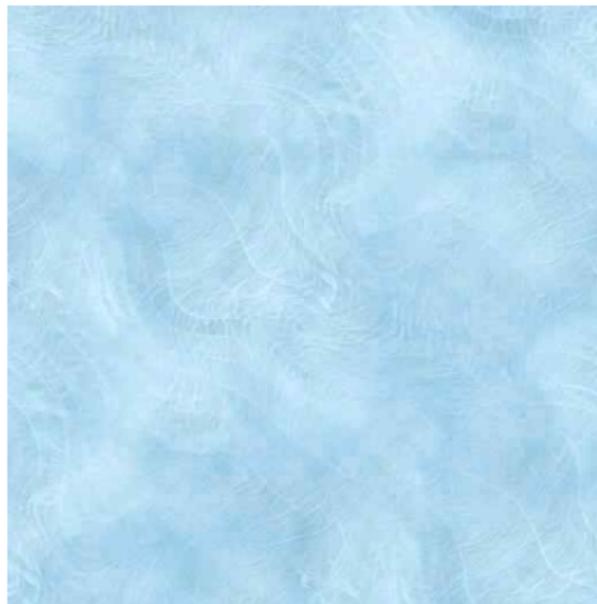
❑ JPEG 종료 시그니처 (Footer):
FF D9 (Ctrl + F 버튼으로 검색 후, 16진수 탭 선택 > FF D8 검색)





해당 .jpg 파일 열어보기

☐ 잘 복원되었습니다





미션

- ❑ <https://creamerburger.tistory.com/72>
- ❑ 해당 사이트 하단에 있는 ybo.pcapng 파일에서 용보공과 관련된 포스터를 추출해보세요. 가운데 있는 사람이 누군지 맞춰보세요!
- ❑ 힌트 : JPG 말고 다른 이미지 형식입니다.

