

最近、パケットのようすが
ちょっとおかしいんだが。

CTF
for
ビギナーズ
ネットワーク講習

※タイトルと内容は関係ないかもしれません

自己紹介

- 保要 隆明 (ほよう たかあき)
- 理工系大学院 修士課程1年
- CTF team : *****
- twitter : @takahoyo (ほよたか)
- 好きなツール : **Wireshark, Nmap**



本日の内容

- CTFの問題【ネットワーク】
- パケット・通信プロトコルについて
- Wiresharkの使い方
- 今後のレベルアップするためには

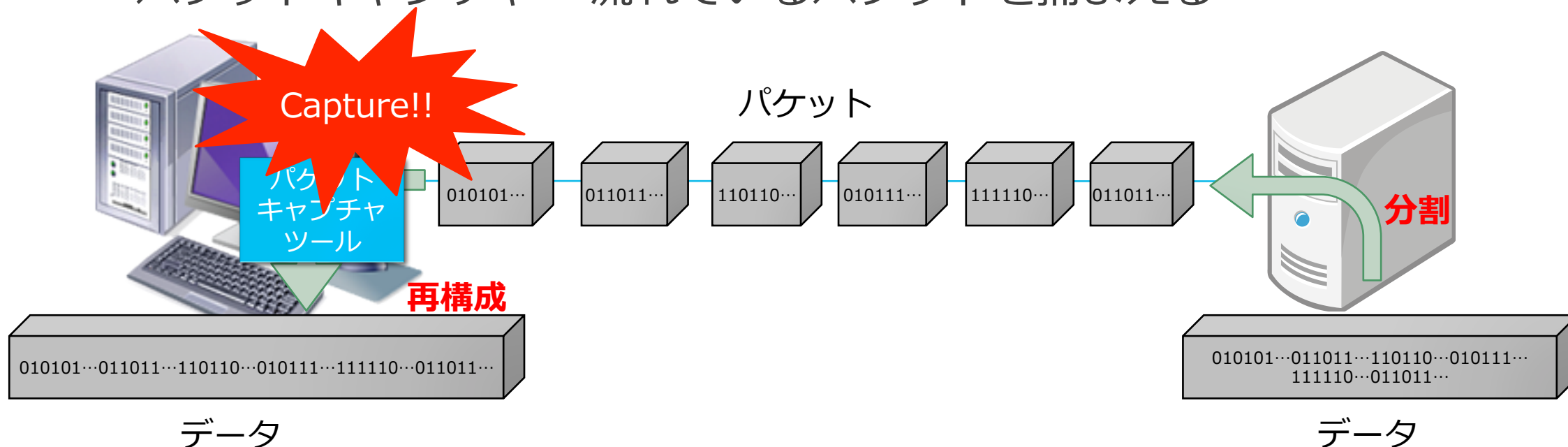
CTFの問題【ネットワーク】

- pcap(pcap-ng)ファイルが与えられる問題
 - ネットワークパケット
 - USBパケット
 - これらのパケットを解析する
 - 解析して、時には解析した情報からサーバへアクセス
- サーバにアクセスして通信を調べる問題
 - パケットキャプチャする
 - 普通にアクセスするだけでなくパケットの中身进行操作することも

ネットワークのプロトコルに関する知識が重要！！

パケット(packet)とは

- 直訳 → 小包、小箱の意味
- ネットワーク上には分割されたデータ = **パケット**が流れている
- パケットキャプチャ = 流れているパケットを捕まえる



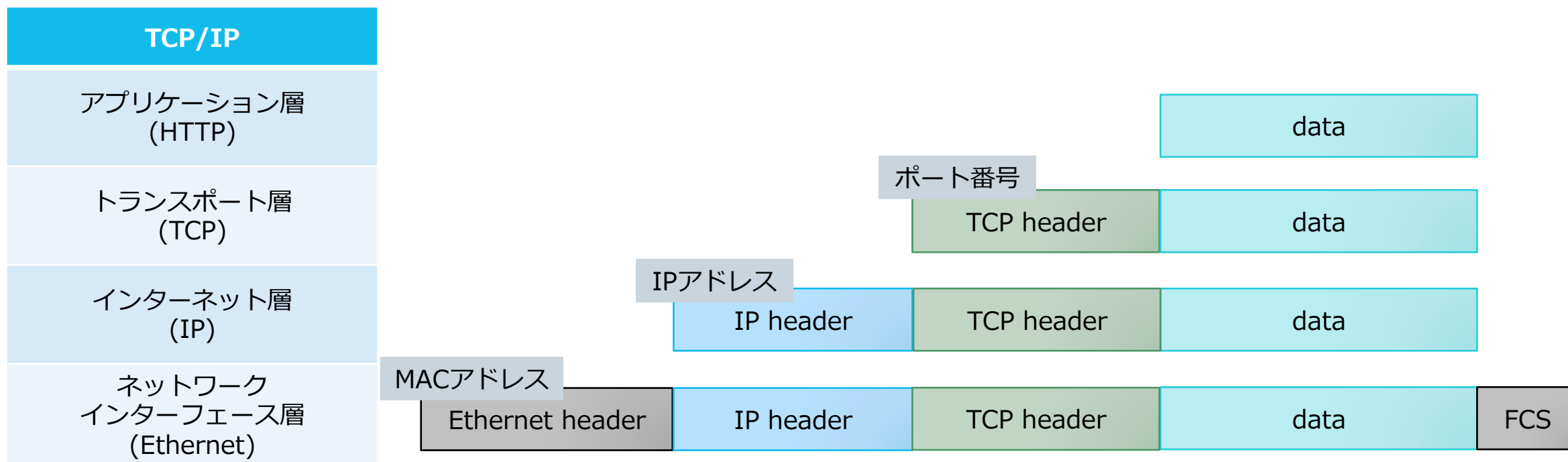
通信プロトコル

- 通信は、プロトコルという決まりによって行われる
- 通信プロトコルは、役割ごとに階層が分かれている

OSI参照モデル	TCP/IP	主なプロトコル
アプリケーション層	アプリケーション層	HTTP, FTP, SMTP, POP3 TELNET, SSH, DNS
プレゼンテーション層		
セッション層		
トランスポート層	トランスポート層	TCP, UDP
ネットワーク層	インターネット層	IP, ICMP
データリンク層	ネットワーク インターフェース層	Ethernet, ARP
物理層		

通信プロトコルとパケットの構造

- 各プロトコルの階層ごとに制御情報(Header)が付与



pcap(pcap-ng)ファイル？

- キャプチャしたパケットを記録したファイル
- fileコマンドすると...

```
$ file example.pcap
example.pcap: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture length 65535)
```

```
$ file example.pcapng
example.pcapng: pcap-ng capture file - version 1.0
```

※ Macのfileコマンドだと、pcap-ngファイルはdataとして扱われる

pcap(pcap-ng)ファイル？

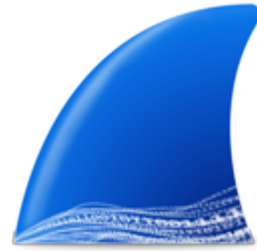
- バイナリエディタで見ると…

example.pcap																	0123456789ABCDEF
ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	D4	C3	B2	A1	02	00	04	00	00	00	00	00	00	00	00	00	ヤテイ.....
00000010	FF	FF	00	00	01	00	00	00	2D	F6	6C	53	B2	A7	09	00-・Sア..
00000020	4A	00	00	00	4A	00	00	00	0C	29	4A	52	68	00	50		J...J.....)JRh.P
00000030	56	C0	00	02	08	00	45	00	00	3C	18	4A	40	00	40	06	Vタ....E...<.J@.@.
00000040	CA	2B	AC	11	00	01	AC	11	00	23	A7	65	00	50	7E	30	ハ+ヤ...ヤ...#アe.P^0
00000050	CB	35	00	00	00	00	A0	02	16	D0	F0	FF	00	00	02	04	ヒ5.....ミ.....
00000060	05	B4	04	02	08	0A	23	4E	D3	7F	00	00	00	00	01	03	.エ....#Nモ.....
00000070	03	07	2D	F6	6C	53	AE	A8	09	00	4A	00	00	00	4A	00	..-・Sヨイ..J...J.
00000080	00	00	00	50	56	C0	00	02	00	0C	29	4A	52	68	08	00	...PVタ....)JRh..
00000090	45	00	00	3C	00	00	40	00	40	06	E2	75	AC	11	00	23	E...<..@.@.穹ヤ..#
000000A0	AC	11	00	01	00	50	A7	65	C3	A0	AC	1C	7E	30	CB	36	ヤ....Pアeア.ヤ..0ヒ6
000000B0	A0	12	16	A0	3D	53	00	00	02	04	05	B4	04	02	08	0A=S.....I....
000000C0	C8	CF	7B	43	23	4E	D3	7F	01	03	03	02	2D	F6	6C	53	ネ[C#Nモ.....-・S
000000D0	CB	A8	09	00	42	00	00	00	42	00	00	00	00	0C	29	4A	ヒ...B...B.....)J
000000E0	52	68	00	50	56	C0	00	02	08	00	45	00	00	34	18	4B	Rh.PVタ....E...4.K
000000F0	40	00	40	06	CA	32	AC	11	00	01	AC	11	00	23	A7	65	@.@./ア2ヤ...ヤ...#アe

example.pcapng																	0123456789ABCDEF
ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	0A	0D	0D	0A	74	00	00	00	4D	3C	2B	1A	01	00	00	00t...M<+.....
00000010	FF	FF	FF	FF	FF	FF	FF	FF	03	00	1C	00	36	34	2D	6264-b
00000020	69	74	20	57	69	6E	64	6F	77	73	20	38	2C	20	62	75	it Windows 8, bu
00000030	69	6C	64	20	39	32	30	30	04	00	2F	00	44	75	6D	70	ild 9200../.Dump
00000040	63	61	70	20	31	2E	31	30	2E	30	20	28	53	56	4E	20	cap 1.10.0 (SYN
00000050	52	65	76	20	34	39	37	39	30	20	66	72	6F	6D	20	2F	Rev 49790 from /
00000060	74	72	75	6E	6B	2D	31	2E	31	30	29	00	00	00	00	00	trunk-1.10).....
00000070	74	00	00	00	01	00	00	00	78	00	00	00	01	00	00	00	t.....x.....
00000080	FF	FF	00	00	02	00	32	00	5C	44	65	76	69	63	65	5C2.¥Device¥
00000090	4E	50	46	5F	7B	44	30	32	42	32	30	34	42	2D	42	43	NPF_{D02B204B-BC
000000A0	43	39	2D	34	30	41	43	2D	41	35	38	35	2D	35	46	38	C9-40AC-A585-5F8
000000B0	44	36	37	31	46	38	31	30	46	7D	00	00	09	00	01	00	D671F810F}.....
000000C0	06	00	00	00	0C	00	1C	00	36	34	2D	62	69	74	20	5764-bit W
000000D0	69	6E	64	6F	77	73	20	38	2C	20	62	75	69	6C	64	20	indows 8, build
000000E0	39	32	30	30	00	00	00	00	78	00	00	00	04	00	00	00	9200....x.....
000000F0	1C	C7	00	00	01	00	17	00	73	A5	9F	C9	61	31	35	38	.又.....s・淤a158

pcapファイルを見るためには…

- ネットワーク解析ツールを利用する
- 主な解析ツール
 - **Wireshark** (Windows, OS X, Linux)
 - **Network Miner** (Windows)
- 今回はWiresharkの使い方を主に説明

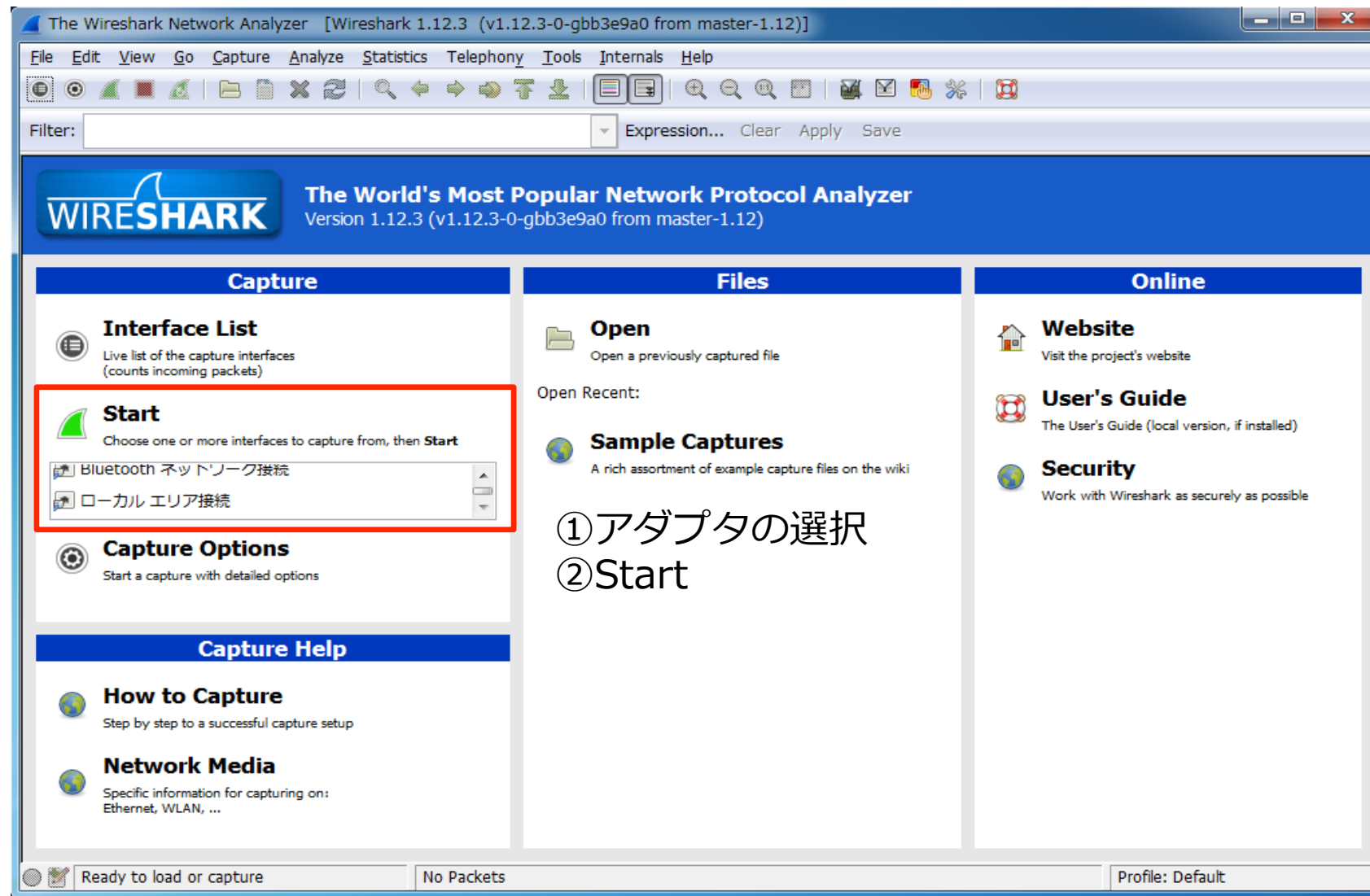


Wireshark

Wiresharkの主な機能

- ネットワークを流れるパケットをキャプチャ
- キャプチャしたパケットを表示する
- 表示したパケットを解析する
 - 指定した条件でフィルタリング
 - パケットからファイルを抽出
 - TCPやUDPのデータ部分(ペイロード)を取り出す
 - 通信を行ってるIPアドレスの統計を表示
 - その他にもいろいろ（紹介したらキリがない）

パケットキャプチャ機能



- ①アダプタの選択
- ②Start

詳細なパケット表示機能

Display Filter

ディスプレイフィルタ

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.75.133	192.168.75.2	DNS	74	Standard query 0x552c A 2014.secon.jp
2	0.018790	192.168.75.2	192.168.75.133	DNS	174	Standard query response 0x552c A 133.242.50.254
3	0.019705	192.168.75.133	133.242.50.254	TCP	66	1513→80 [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=256 SACK
4	0.050884	133.242.50.254	192.168.75.133	TCP	60	80→1513 [SYN, ACK] Seq=0 Ack=1 win=64240 Len=0 MSS=1460
5	0.050961	192.168.75.133	133.242.50.254	TCP	54	1513→80 [ACK] Seq=1 Ack=1 win=64240 Len=0
6	0.051324	192.168.75.133	133.242.50.254	HTTP	401	GET / HTTP/1.1
7	0.051513	133.242.50.254	192.168.75.133	TCP	60	80→1513 [ACK] Seq=1 Ack=348 win=64240 Len=0
8	0.355343	133.242.50.254	192.168.75.133	TCP	249	[TCP segment of a reassembled PDU]
9	0.355456	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
10	0.355456	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
11	0.355459	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
12	0.355460	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
13	0.355460	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

- Ethernet II, Src: Vmware_bf:22:b4 (00:0c:29:bf:22:b4), Dst: Vmware_f4:81:5f (00:50:56:f4:81:5f)
- Internet Protocol Version 4, Src: 192.168.75.133 (192.168.75.133), Dst: 192.168.75.2 (192.168.75.2)
- User Datagram Protocol, Src Port: 61902 (61902), Dst Port: 53 (53)
- Domain Name System (query)

0000 00 50 56 f4 81 5f 00 0c 29 bf 22 b4 08 00 45 00 .PV....)."...E.
0010 00 3c 1c 3c 00 00 80 11 00 00 c0 a8 4b 85 c0 a8 .<.<....K...
0020 4b 02 f1 ce 00 35 00 28 18 12 55 2c 01 00 00 01 K....5.(..U,....
0030 00 00 00 00 00 00 04 32 30 31 34 06 73 65 63 632 014.secc
0040 6f 6e 02 6a 70 00 00 01 00 01 on.jp... ..

File: "\\vmware-host\Shared Folders\ドキ... Packets: 466 · Displayed: 466 (100.0%) · Load time: 0:00.062 Profile: Default

Packet List

パケットの一覧

Packet Details

パケットの詳細

Packet Bytes

生のパケット

Packet List (パケット一覧)

時間

送信元アドレス

送信先アドレス

プロトコル

パケット長

パケットの概要

View->NameResolution
の"Enable for Transport Layer"
のチェックを外す

1513→80 [SYN]

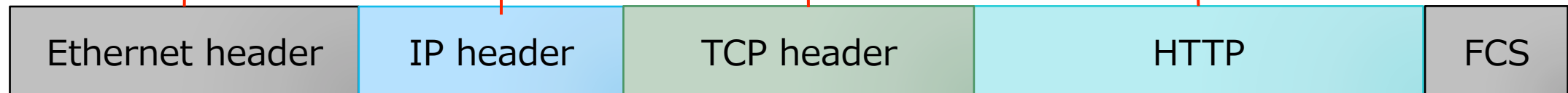
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.75.133	192.168.75.2	DNS	74	Standard query 0x552c A 2014.secon.jp
2	0.018790	192.168.75.2	192.168.75.133	DNS	174	Standard query response 0x552c A 133.242.50.254
3	0.019705	192.168.75.133	133.242.50.254	TCP	66	fujitsu-dtc-http [SYN] Seq=0 win=8192 Len=0 MSS=1460
4	0.050884	133.242.50.254	192.168.75.133	TCP	60	http-fujitsu-dtc [SYN, ACK] Seq=1513 Ack=348 win=64240 Len=0
5	0.050961	192.168.75.133	133.242.50.254	TCP	54	fujitsu-dtc-http [ACK] Seq=1 Ack=348 win=0 Len=0
6	0.051324	192.168.75.133	133.242.50.254	HTTP	401	GET / HTTP/1.1
7	0.051513	133.242.50.254	192.168.75.133	TCP	60	http-fujitsu-dtc [ACK] Seq=1 Ack=348 win=64240 Len=0
8	0.355343	133.242.50.254	192.168.75.133	TCP	249	[TCP segment of a reassembled PDU]
9	0.355456	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
10	0.355456	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
11	0.355459	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
12	0.355460	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]
13	0.355460	133.242.50.254	192.168.75.133	TCP	1514	[TCP segment of a reassembled PDU]

Edit->Perference->User Interfaces ->Columns
でカスタマイズ可

View->Time Display Format
で形式を変更可

Packet Details (パケットの詳細)

```
+ Frame 6: 401 bytes on wire (3208 bits), 401 bytes captured (3208 bits) on interface 0
+ Ethernet II, Src: Vmware_bf:22:b4 (00:0c:29:bf:22:b4), Dst: Vmware_f4:81:5f (00:50:56:f4:81:5f)
+ Internet Protocol Version 4, Src: 192.168.75.133 (192.168.75.133), Dst: 133.242.50.254 (133.242.50.254)
+ Transmission Control Protocol, Src Port: 1513 (1513), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 347
- Hypertext Transfer Protocol
  + GET / HTTP/1.1\r\n
    Host: 2014.seccon.jp\r\n
    User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: ja,en-us;q=0.7,en;q=0.3\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    If-Modified-Since: Fri, 20 Jun 2014 10:33:09 GMT\r\n
    \r\n
    [Full request URI: http://2014.seccon.jp/]
```



Packet Bytes (生のパケット)

- 通常の人間には読みづらい形式(16進数)
- Wiresharkが解析し、整形してくれている

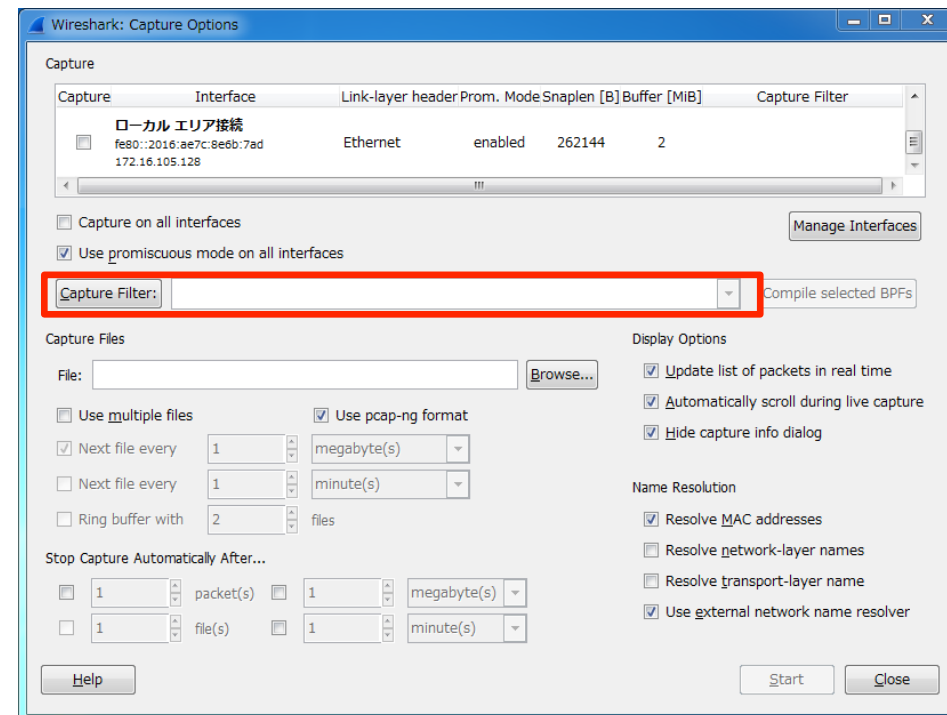
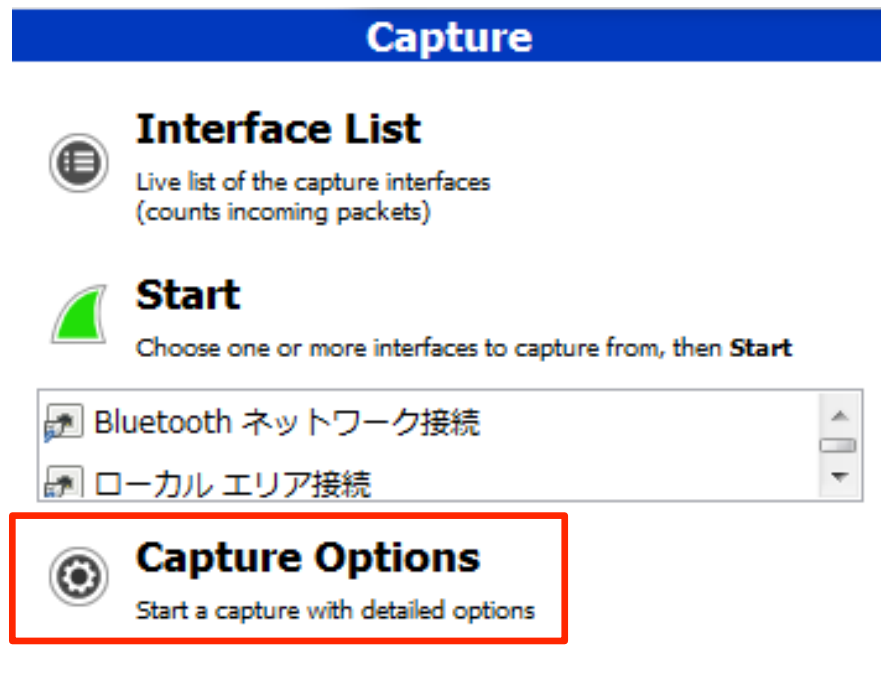
0000	00	50	56	f4	81	5f	00	0c	29	bf	22	b4	08	00	45	00	.PV..._...). "...E.
0010	01	23	77	75	40	00	80	06	00	00	c0	a8	4b	85	85	f2	.#wu@...K...
0020	32	fe	05	19	00	50	ce	68	b9	57	2e	39	4e	76	50	18	2....P.h .w.9NvP.
0030	fa	f0	c6	33	00	00	47	45	54	20	2f	20	48	54	54	50	...3..GE T / HTTP
0040	2f	31	2e	31	0d	0a	41	63	63	65	70	74	3a	20	74	65	/1.1..Ac cept: te
0050	78	74	2f	68	74	6d	6c	2c	20	61	70	70	6c	69	63	61	xt/html, applica
0060	74	69	6f	6e	2f	78	68	74	6d	6c	2b	78	6d	6c	2c	20	tion/xht ml+xml,
0070	2a	2f	2a	0d	0a	41	63	63	65	70	74	2d	4c	61	6e	67	*/*..Acc ept-Lang
0080	75	61	67	65	3a	20	6a	61	2d	4a	50	0d	0a	55	73	65	uage: ja -JP..Use
0090	72	2d	41	67	65	6e	74	3a	20	4d	6f	7a	69	6c	6c	61	r-Agent: Mozilla
00a0	2f	35	2e	30	20	28	63	6f	6d	70	61	74	69	62	6c	65	/5.0 (co mpatible
00b0	3b	20	4d	53	49	45	20	39	2e	30	3b	20	57	69	6e	64	; MSIE 9 .0; wind
00c0	6f	77	73	20	4e	54	20	36	2e	31	3b	20	57	4f	57	36	ows NT 6 .1; WOW6
00d0	34	3b	20	54	72	69	64	65	6e	74	2f	35	2e	30	29	0d	4; Tride nt/5.0).
00e0	0a	41	63	63	65	70	74	2d	45	6e	63	6f	64	69	6e	67	.Accept- Encoding
00f0	3a	20	67	7a	69	70	2c	20	64	65	66	6c	61	74	65	0d	: gzip, deflate.
0100	0a	48	6f	73	74	3a	20	32	30	31	34	2e	73	65	63	63	.Host: 2 014.secc

【目的別】 Wiresharkの使い方

- 条件にあった通信を抜き出したい → **Filter**
- TCPが送信されるデータを抜きたい → **Follow TCP Streams**
- HTTPで扱ってるファイルを抽出 → **Export Object->HTTP**
- パケットから生データ抽出 → **Export Selected Packet Bytes**
- パケットの様々な統計を知りたい → **Statistics**

Filter

- Capture Filter



- Display Filter



Filter

- Capture Filter
 - パケットキャプチャを始める前に指定
 - キャプチャしたいパケットが決まっている時に有効
 - 無駄なパケットをとらなくて済む
- Display Filter
 - パケット表示画面で指定(パケットキャプチャしながら指定可能)
 - キャプチャ中もキャプチャ後もFilterをかけることが可能
- 各フィルタで書式が違う！！

Display Filter

- プロトコルの指定
- 例
 - IPを使ってるパケットのみ
 - ICMPを使ってるパケットのみ
 - TCPを使ってるパケットのみ
 - UDPを使ってるパケットのみ
 - HTTPを使ってるパケットのみ

Filter: ip

Filter: icmp

Filter: tcp

Filter: udp

Filter: http

Display Filter

- プロトコルの要素でフィルタリング
- 例
- TCPの80番ポートを利用している通信をフィルタ
 - `tcp.port == 80`
- IPアドレスが133.242.50.254の通信をフィルタ
 - `ip.addr == 133.242.50.254`
- IPアドレスが133.242.50.254で、TCP80番ポートの通信
 - `ip.addr == 133.242.50.254 && tcp.port == 80`
- TCPでSYNフラグが立っているパケットをフィルタ
 - `tcp.flags.syn == 1`

Display Filter

よく使う（独断と偏見）フィルタの構文

フィルタ	意味
ip.addr == IPアドレス	IPアドレス
ip.src == IPアドレス	送信元のIPアドレス
ip.dst == IPアドレス	送信先のIPアドレス
tcp.flags == 0x02	TCPパケット(syn)
tcp.flags == 0x12	TCPパケット(syn/ack)
tcp.flags == 0x14	TCPパケット(rst/ack)
tcp.port == ポート番号	TCPのポート番号
tcp.srcport == ポート番号	TCPの送信元ポート番号
tcp.dstport == ポート番号	TCPの送信先ポート番号
http.request	HTTPのリクエスト
http.responce	HTTPのレスポンス

比較演算子

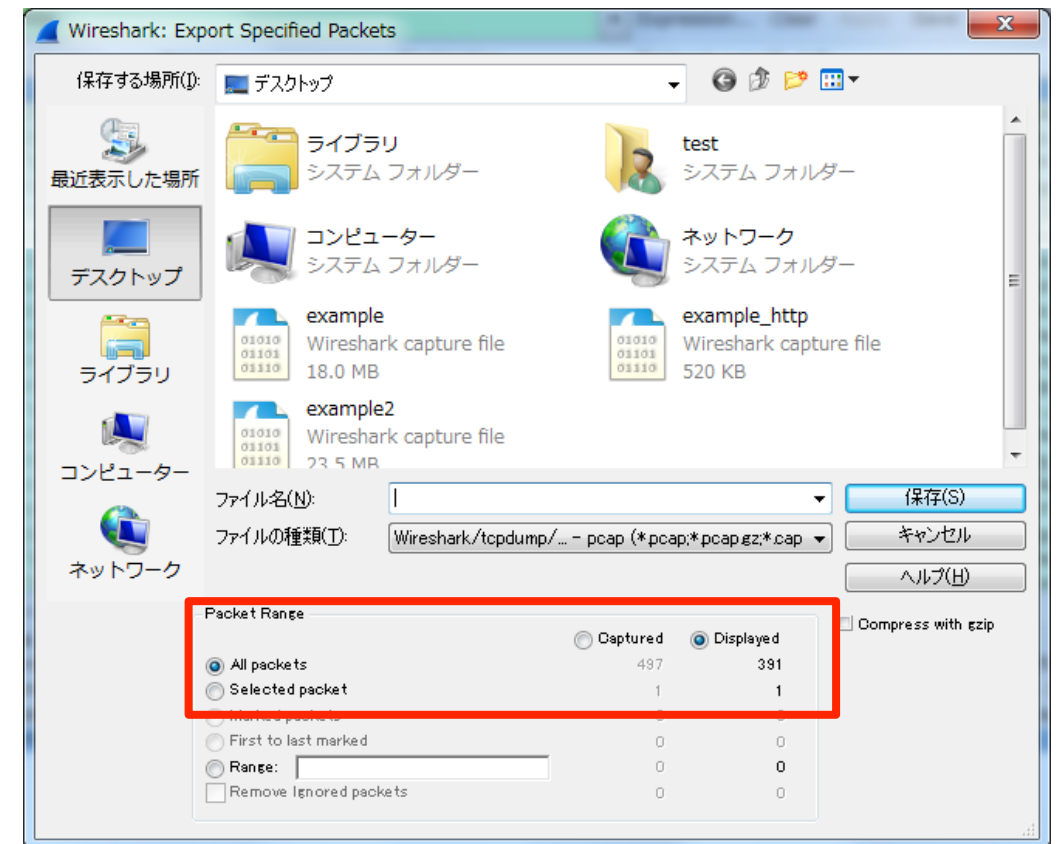
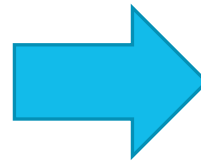
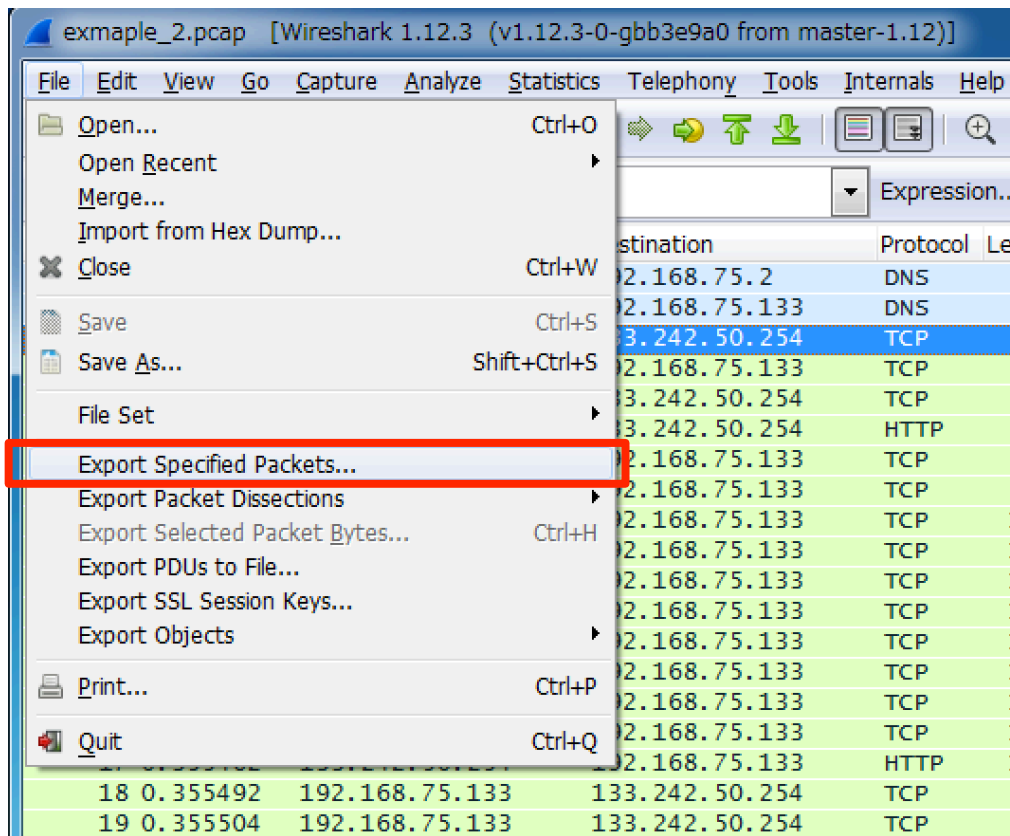
比較演算子	意味
eq (==)	等しい
ne (!=)	等しくない
gt (>)	大きい
lt (<)	小さい
ge (>=)	以上
le (<=)	以下

論理演算子

論理演算子	意味
and (&&)	論理積（かつ）
or ()	論理和（または）
not (!)	否定

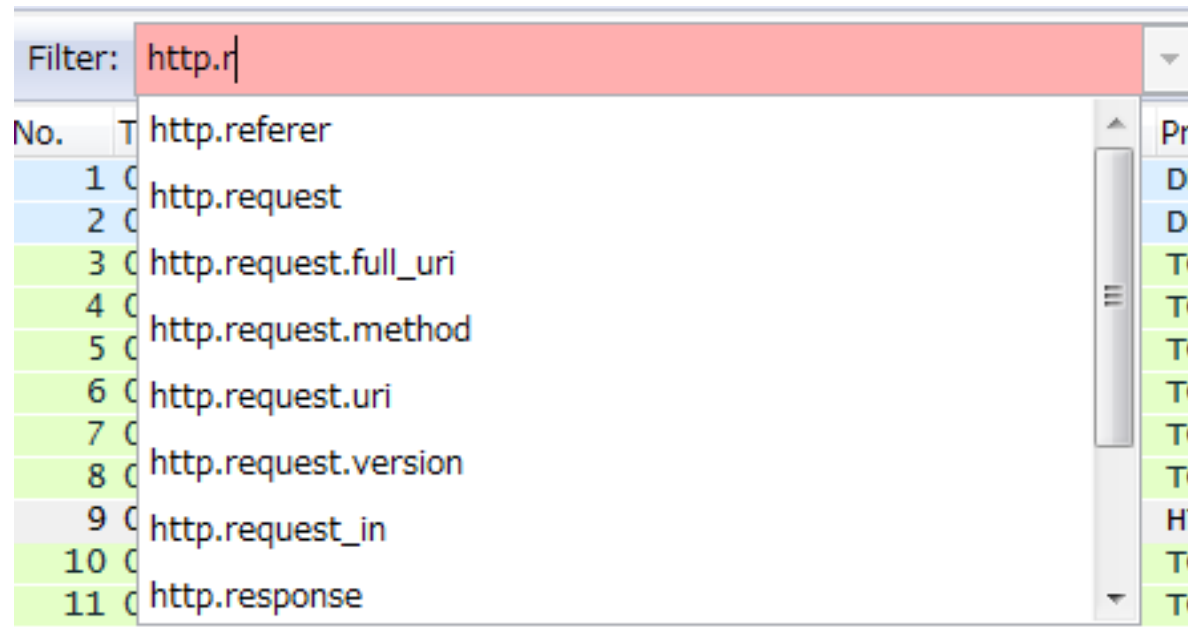
Display Filter

- Display Filterしたパケットを保存することも可能
- File -> Export Specified Packets



Display Filter

- フィルタの構文忘れた → 補完機能
 - 構文が間違っていると、ウィンドウが赤くなる
 - 絞りたい情報があるけど、構文がわからん
- 補完機能でそれらしき構文を選んでみて、試してみる

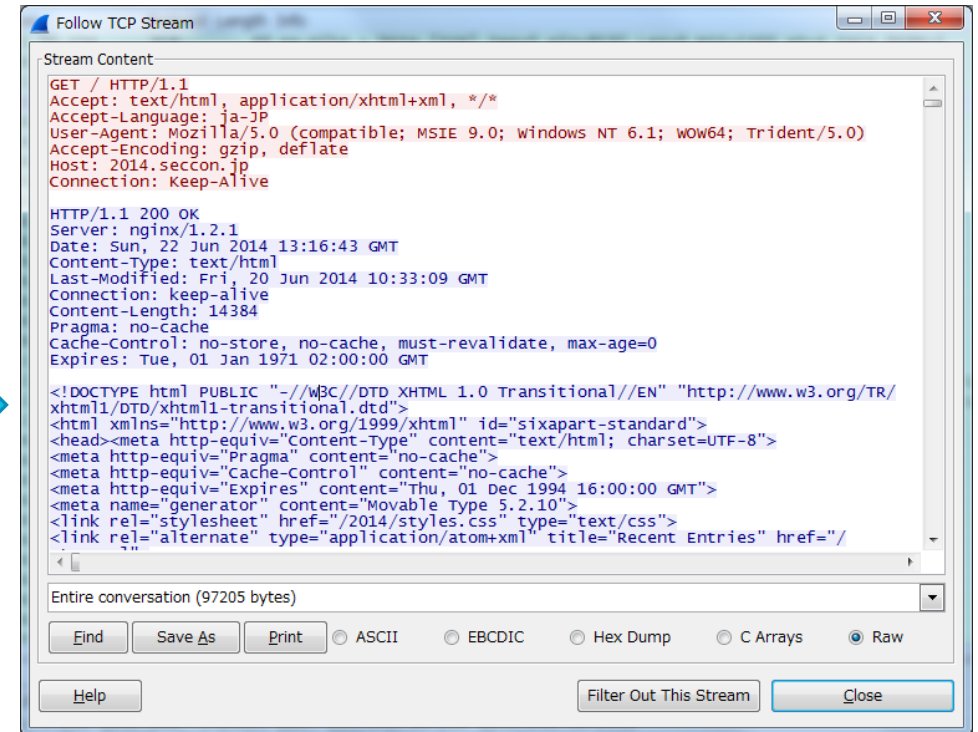
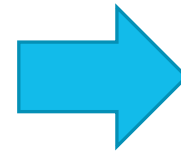


TCPで送信されるデータを抜きたい

- TCPパケットを選択し、右クリック -> Follow TCP Stream

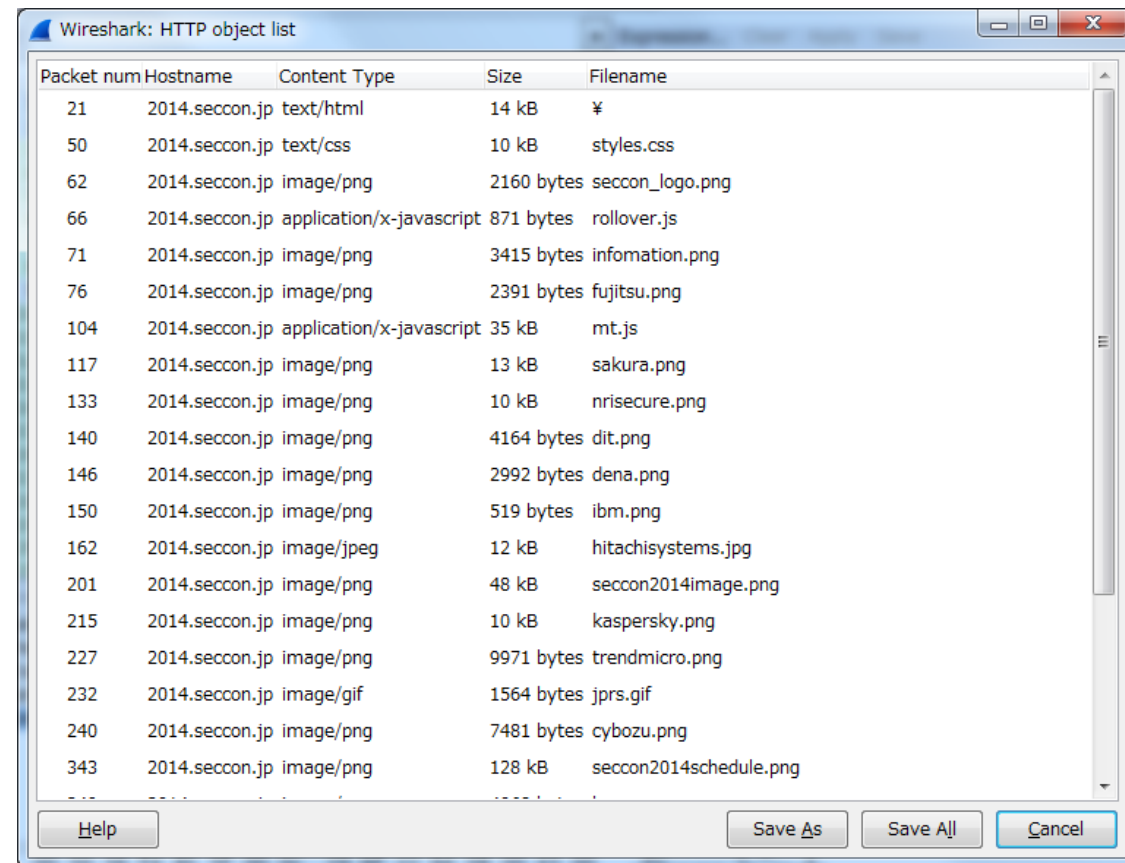
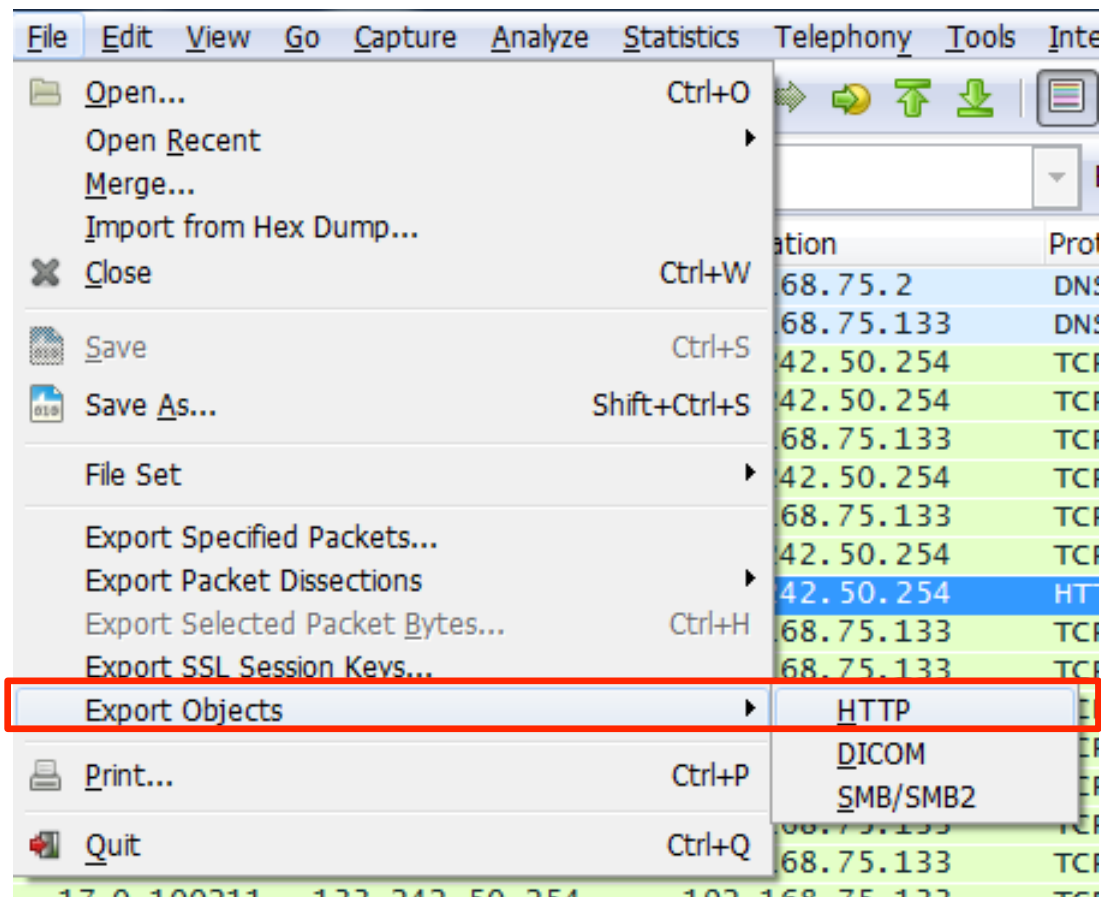
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.75.133	192.168.75.2	DNS	74	Standard query 0x552c
2	0.018790	192.168.75.2	192.168.75.133	DNS	174	Standard query response
3	0.019705	192.168.75.133	133.242.50.254	TCP		
4	0.050884	133.242.50.254	192.168.75.133	TCP		
5	0.050961	192.168.75.133	133.242.50.254	TCP		
6	0.051324	192.168.75.133	133.242.50.254	HTTP		
7	0.051513	133.242.50.254	192.168.75.133	TCP		
8	0.355343	133.242.50.254	192.168.75.133	TCP		
9	0.355456	133.242.50.254	192.168.75.133	TCP		
10	0.355456	133.242.50.254	192.168.75.133	TCP		
11	0.355459	133.242.50.254	192.168.75.133	TCP		
12	0.355460	133.242.50.254	192.168.75.133	TCP		
13	0.355460	133.242.50.254	192.168.75.133	TCP		
14	0.355461	133.242.50.254	192.168.75.133	TCP		
15	0.355461	133.242.50.254	192.168.75.133	TCP		
16	0.355462	133.242.50.254	192.168.75.133	TCP		
17	0.355462	133.242.50.254	192.168.75.133	HTTP		
18	0.355492	192.168.75.133	133.242.50.254	TCP		
19	0.355504	192.168.75.133	133.242.50.254	TCP		
20	0.405457	192.168.75.133	133.242.50.254	HTTP		
21	0.405604	133.242.50.254	192.168.75.133	TCP		
22	0.406041	192.168.75.133	133.242.50.254	TCP		
23	0.406264	192.168.75.133	133.242.50.254	TCP		

Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: Vmware_bf:22:b4 (00:0c:29:bf:22:b4), Dst: 192.168.75.133 (08:00:27:00:00:00)
Internet Protocol Version 4, Src: 192.168.75.133, Dst: 133.242.50.254
Transmission Control Protocol, Src Port: 1513 (1513), Dst Port: 80 (80)



HTTPで取り扱ってるファイルを抽出

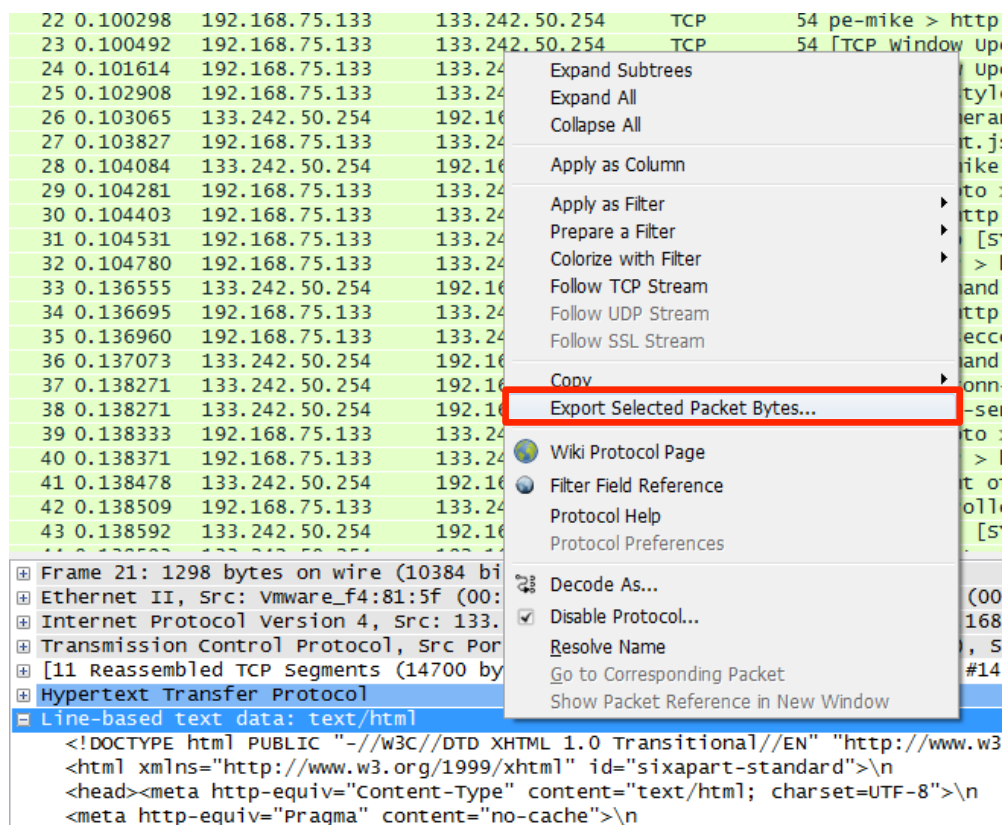
- File -> Export Objects -> HTTP



パケットから生データを抽出

- パケットの詳細画面で、抽出したい部分を右クリック

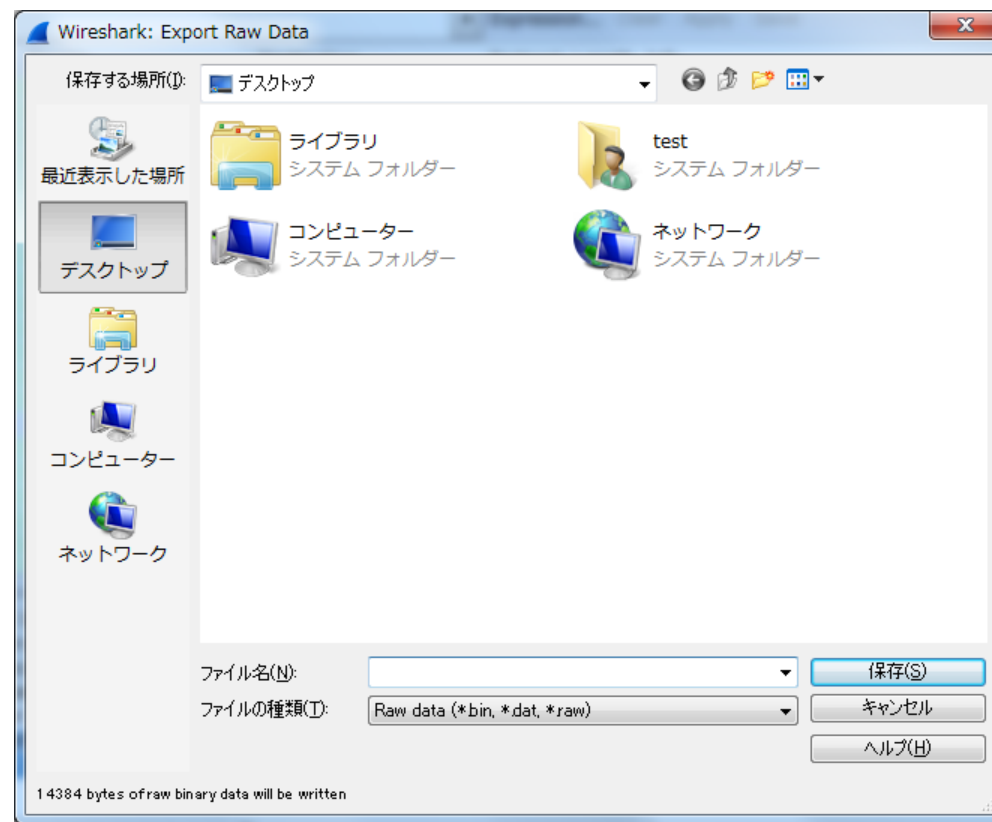
->Export Selected Packet Bytes



The image shows the Wireshark interface. The packet list pane on the left displays a list of captured packets. The details pane on the right shows the selected packet (Frame 21) and its structure. A right-click context menu is open over the packet list, with the option "Export Selected Packet Bytes..." highlighted in red. The menu also includes options like "Expand Subtrees", "Apply as Filter", "Copy", and "Decode As...".

No.	Time	Source	Destination	Protocol	Length	Info
22	0.100298	192.168.75.133	133.242.50.254	TCP	54	pe-mike > http
23	0.100492	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
24	0.101614	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
25	0.102908	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
26	0.103065	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
27	0.103827	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
28	0.104084	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
29	0.104281	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
30	0.104403	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
31	0.104531	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
32	0.104780	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
33	0.136555	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
34	0.136695	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
35	0.136960	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
36	0.137073	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
37	0.138271	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
38	0.138271	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
39	0.138333	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
40	0.138371	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
41	0.138478	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]
42	0.138509	192.168.75.133	133.242.50.254	TCP	54	[TCP window up]
43	0.138592	133.242.50.254	192.168.75.133	TCP	54	[TCP window up]

Frame 21: 1298 bytes on wire (10384 bits) captured on interface eth0
Ethernet II, Src: Vmware_f4:81:5f (00:0c:29:f4:81:5f), Dst: 133.242.50.254 (01:00:0c:29:f4:81:5f)
Internet Protocol Version 4, Src: 192.168.75.133, Destination: 133.242.50.254
Transmission Control Protocol, Src Port: 5444, Dst Port: 80
[11 Reassembled TCP Segments (14700 bytes) ...]
Hypertext Transfer Protocol
Line-based text data: text/html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" id="sixapart-standard">
<head><meta http-equiv="content-type" content="text/html; charset=UTF-8">
<meta http-equiv="Pragma" content="no-cache">



The image shows the "Wireshark: Export Raw Data" dialog box. The "Save to" field is set to "Desktop". The "Save as" field is empty. The "File type" is set to "Raw data (*.bin, *.dat, *.raw)". The "Save" button is highlighted. The dialog also shows a list of recent locations and a preview of the raw data.

保存する場所(D): デスクトップ

最近表示した場所

- デスクトップ
- ライブラリ
- コンピュータ
- ネットワーク

test システム フォルダ

ネットワーク システム フォルダ

ファイル名(N):

ファイルの種類(I): Raw data (*.bin, *.dat, *.raw)

保存(S)

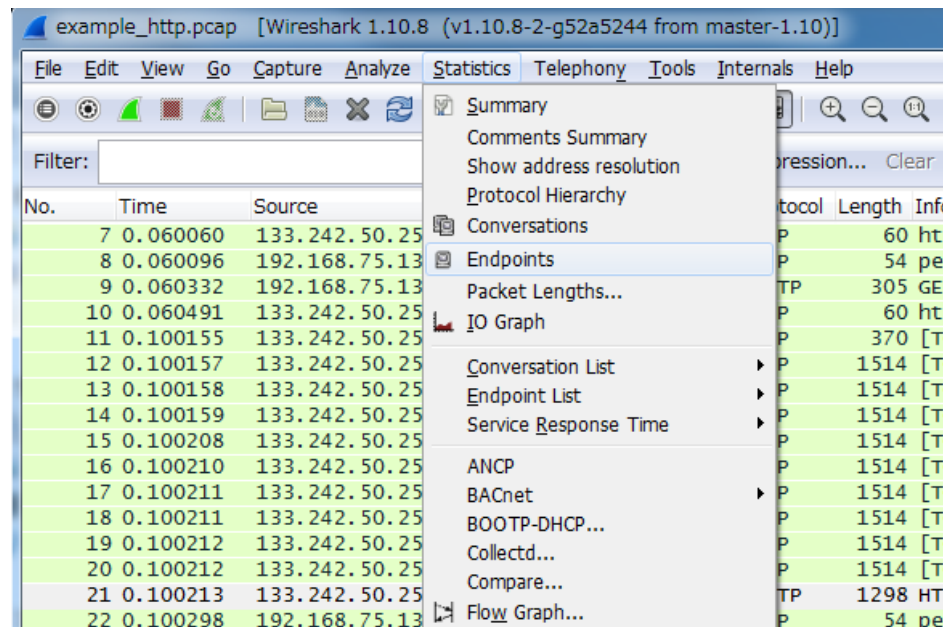
キャンセル

ヘルプ(H)

14384 bytes of raw binary data will be written

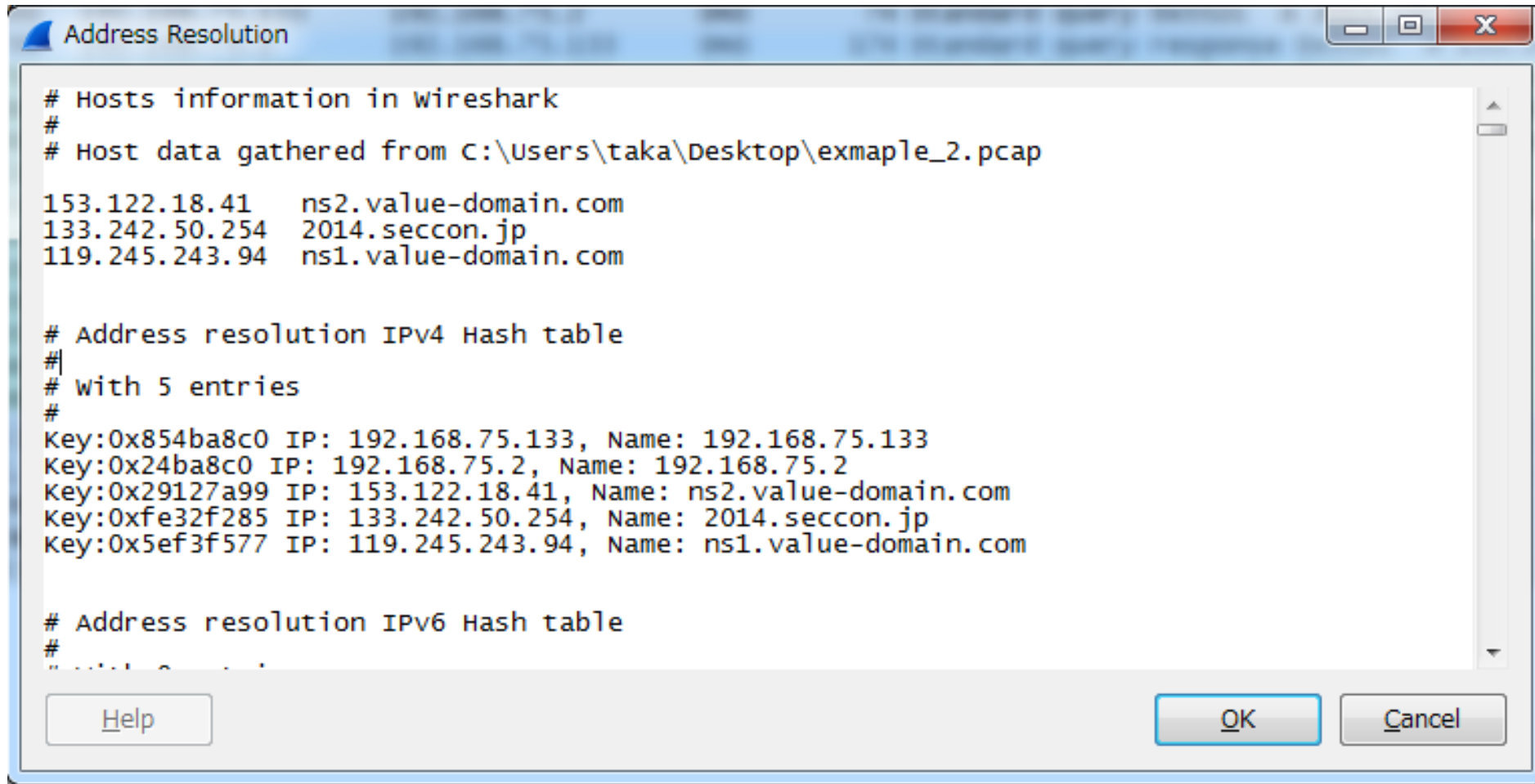
パケットの様々な統計を知りたい

- Statistics
 - IPアドレスとドメイン名を知りたい → Show address resolution
 - 利用されているプロトコルの統計を知りたい → Protocol Hierarchy
 - どの端末がどの端末と通信しているかの統計を知りたい → Conversations
 - どのような端末が通信しているかの統計を知りたい → Endpoints



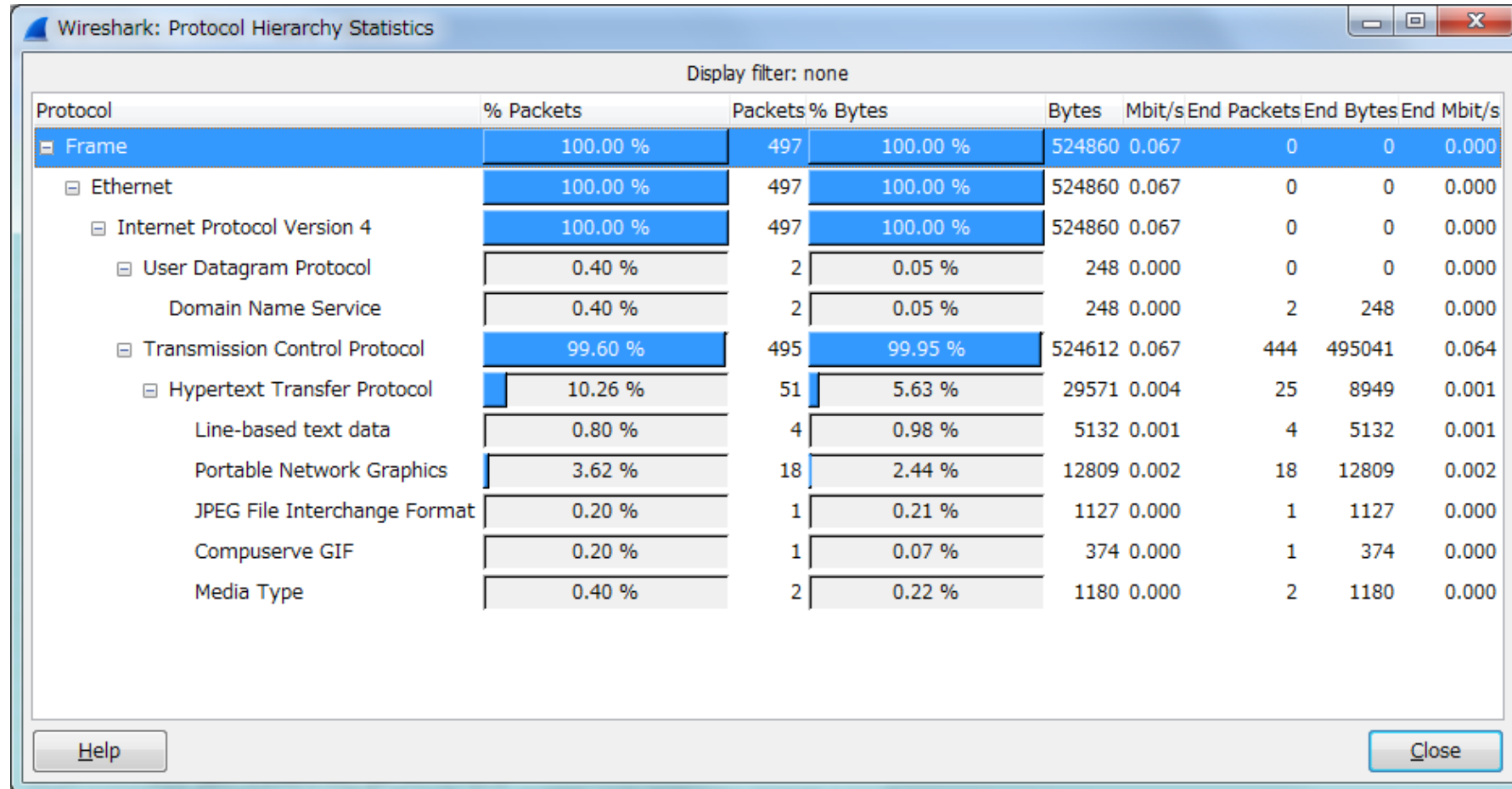
IPアドレスとドメイン名を知りたい

- Statistics -> Show address resolution



利用されているプロトコルの統計を知りたい

- Statistics -> Protocol Hierarchy



Wireshark: Protocol Hierarchy Statistics

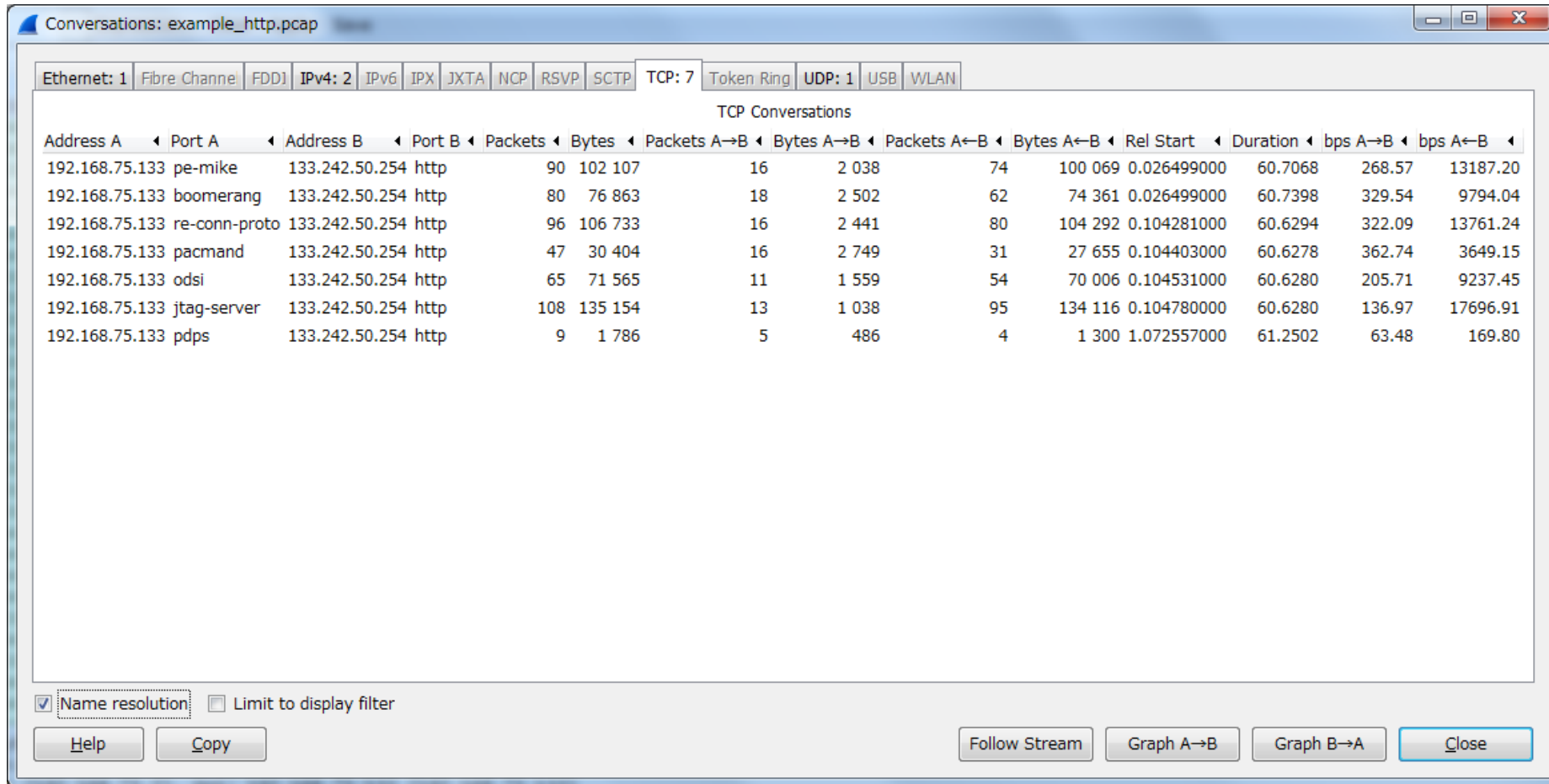
Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	497	100.00 %	524860	0.067	0	0	0.000
Ethernet	100.00 %	497	100.00 %	524860	0.067	0	0	0.000
Internet Protocol Version 4	100.00 %	497	100.00 %	524860	0.067	0	0	0.000
User Datagram Protocol	0.40 %	2	0.05 %	248	0.000	0	0	0.000
Domain Name Service	0.40 %	2	0.05 %	248	0.000	2	248	0.000
Transmission Control Protocol	99.60 %	495	99.95 %	524612	0.067	444	495041	0.064
Hypertext Transfer Protocol	10.26 %	51	5.63 %	29571	0.004	25	8949	0.001
Line-based text data	0.80 %	4	0.98 %	5132	0.001	4	5132	0.001
Portable Network Graphics	3.62 %	18	2.44 %	12809	0.002	18	12809	0.002
JPEG File Interchange Format	0.20 %	1	0.21 %	1127	0.000	1	1127	0.000
Compuserve GIF	0.20 %	1	0.07 %	374	0.000	1	374	0.000
Media Type	0.40 %	2	0.22 %	1180	0.000	2	1180	0.000

Help Close

どの端末がどの端末と通信しているか統計を知りたい

- Statistics -> Conversations

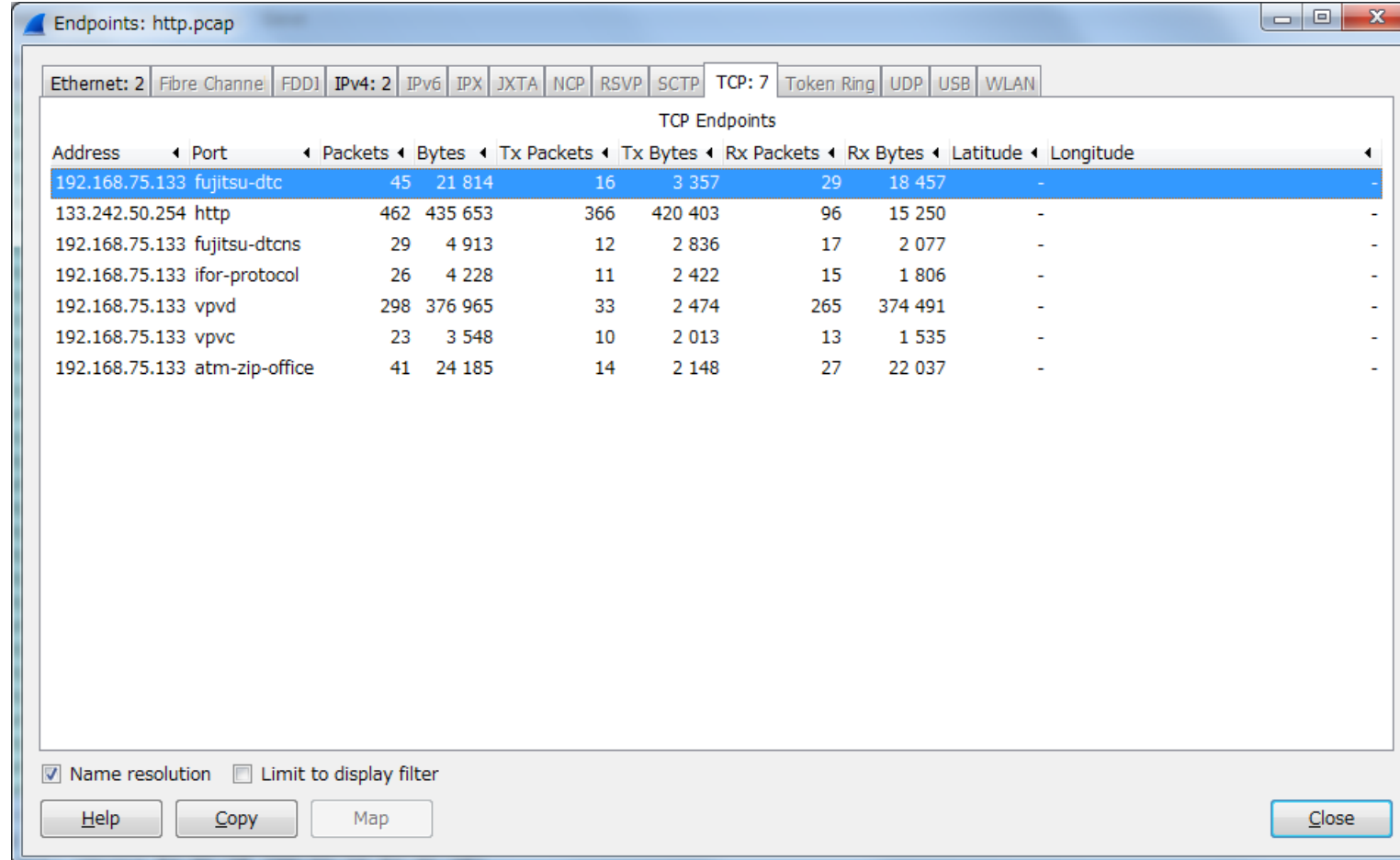


The screenshot shows the 'Conversations' window in Wireshark, filtered for 'example_http.pcap'. The 'TCP' tab is selected, showing a list of 7 TCP conversations. The table below represents the data shown in the window.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
192.168.75.133	pe-mike	133.242.50.254	http	90	102 107	16	2 038	74	100 069	0.026499000	60.7068	268.57	13187.20
192.168.75.133	boomerang	133.242.50.254	http	80	76 863	18	2 502	62	74 361	0.026499000	60.7398	329.54	9794.04
192.168.75.133	re-conn-proto	133.242.50.254	http	96	106 733	16	2 441	80	104 292	0.104281000	60.6294	322.09	13761.24
192.168.75.133	pacmand	133.242.50.254	http	47	30 404	16	2 749	31	27 655	0.104403000	60.6278	362.74	3649.15
192.168.75.133	odsi	133.242.50.254	http	65	71 565	11	1 559	54	70 006	0.104531000	60.6280	205.71	9237.45
192.168.75.133	jtag-server	133.242.50.254	http	108	135 154	13	1 038	95	134 116	0.104780000	60.6280	136.97	17696.91
192.168.75.133	pdps	133.242.50.254	http	9	1 786	5	486	4	1 300	1.072557000	61.2502	63.48	169.80

どのIPアドレスが記録されてるのか知りたい

- Statistics -> Endpoints



Endpoints: http.pcap

Ethernet: 2 | Fibre Channel | FDDI | IPv4: 2 | IPv6 | IPX | JXTA | NCP | RSVP | SCTP | TCP: 7 | Token Ring | UDP | USB | WLAN

TCP Endpoints

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude	Longitude
192.168.75.133	fujitsu-dtc	45	21 814	16	3 357	29	18 457	-	-
133.242.50.254	http	462	435 653	366	420 403	96	15 250	-	-
192.168.75.133	fujitsu-dtcns	29	4 913	12	2 836	17	2 077	-	-
192.168.75.133	ifor-protocol	26	4 228	11	2 422	15	1 806	-	-
192.168.75.133	vpvd	298	376 965	33	2 474	265	374 491	-	-
192.168.75.133	vpvc	23	3 548	10	2 013	13	1 535	-	-
192.168.75.133	atm-zip-office	41	24 185	14	2 148	27	22 037	-	-

☒ Name resolution ☐ Limit to display filter

Help Copy Map Close

ネットワーク系の問題へのアプローチ

- 不審な通信を探し、不審な箇所を探す
 - 通常の通信と不審な通信を見分ける目が必要
 - = ネットワークの知識、パケットを見る経験が必要
 - それを見るための手法
 - = Wireshark等のツールの使い方を極める
- 過去問題の解法を知る = Writeupを読む
- パケットを作れるようになる
 - nmap等を利用してポートスキャン
 - hping, Scapyなどで手動でパケット生成

もっと勉強したい人は…

- 書籍
 - マスタリングTCP/IP 入門編（オーム社）
 - ネットワークの知識を得たい人に
 - 実践パケット解析（オライリー）
 - もっとパケット解析について知りたい人に
- Webサイト
 - 3分間Networking
 - <http://www5e.biglobe.ne.jp/%257eaji/3min/index.html>
 - Wireshark公式サイト(英語)
 - <http://www.wireshark.org/>

Thank You For Listening

Network Packets Don't Lie.

Q&A

付録 1

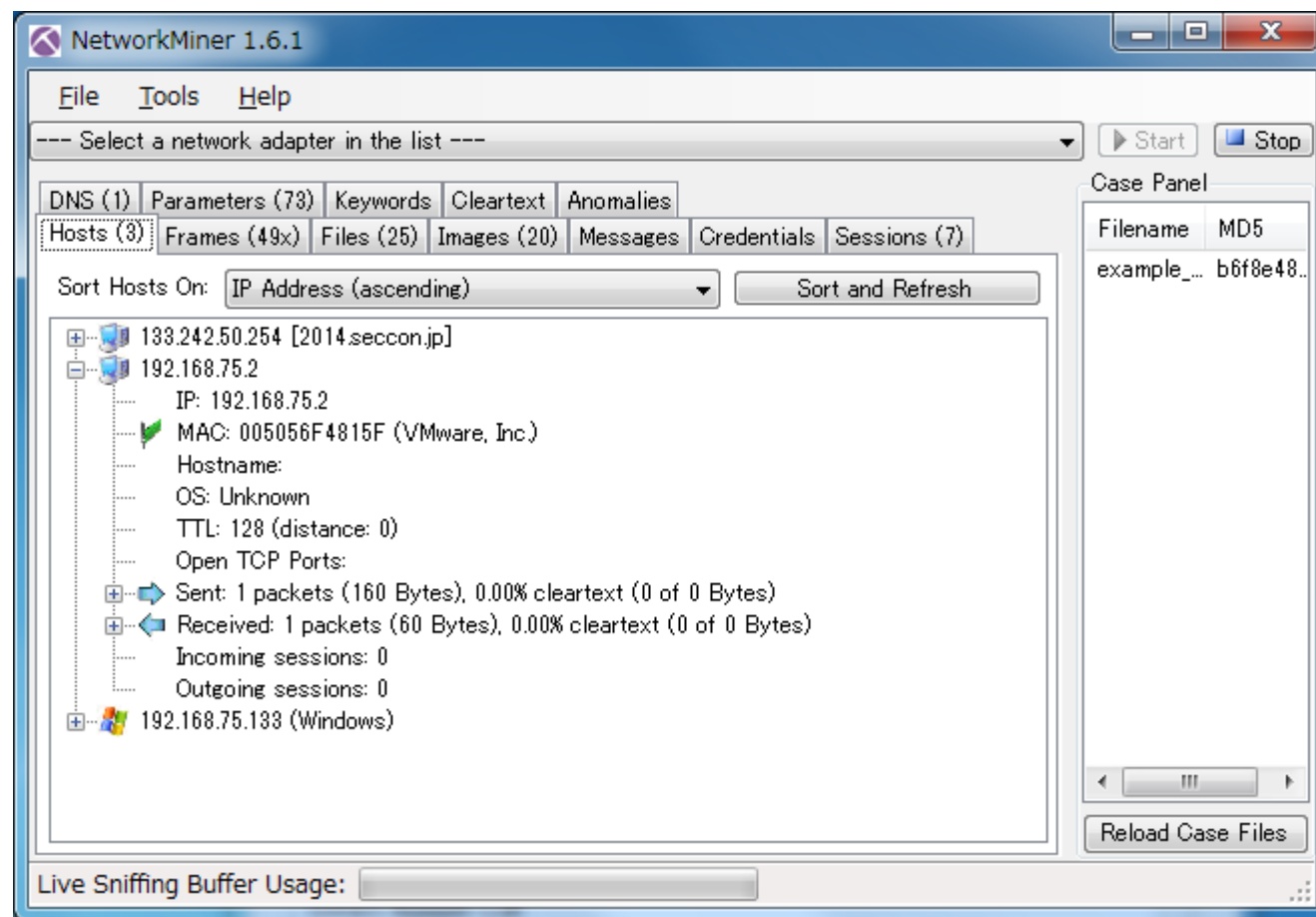
Network Minerの使い方



Network Miner

Network Minerの使い方

pcapファイルをドラッグ&ドロップ



Network Miner

- Network Minerでわかる情報
 - ホスト情報 (OS, Open TCP Port, Service name , version)
 - ファイル抽出
 - 画像抽出 (サムネイル付き)
 - 認証情報
 - etc...

ホスト 情報

OS種類
(passive finger printing)

開いているポート

ホストの詳細

The screenshot shows the NetworkMiner 1.6.1 application window. The main panel displays a list of hosts, with the first host, 192.168.75.132, selected. The host details are expanded, showing various system and network information. The 'Open TCP Ports' section is highlighted, showing ports 80, 111, 3306, 445, 5900, and 21. The 'Host Details' section shows queried DNS names, web server banner, FTP server banner, preferred SMB dialect, SMB native LAN manager, SMB native OS, SSH application, and SSH version.

NetworkMiner 1.6.1

File Tools Help

--- Select a network adapter in the list --- [Start] [Stop]

Parameters (236) Keywords Cleartext Anomalies

Hosts (51) Frames (42xx) Files (18) Images (8) Messages Credentials (4) Sessions (1110) DNS (76)

Sort Hosts On: IP Address (ascending) [Sort and Refresh]

192.168.75.132 [METASPLOITABLE] [192.168.75.132] [nmap] (Linux)

- IP: 192.168.75.132
- MAC: 000C29483A31 (VMware, Inc.)
- Hostname: METASPLOITABLE, 192.168.75.132, nmap
- OS: Linux
- TTL: 64 (distance: 0)
- Open TCP Ports: 80 (Http) 111 3306 445 (NetBiosSessionService) 5900 25 (Smtp) 21 (FtpControl)
- Sent: 1886 packets (268,870 Bytes), 0.00% cleartext (0 of 0 Bytes)
- Received: 1741 packets (92,325 Bytes), 0.00% cleartext (0 of 0 Bytes)
- Incoming sessions: 138
- Outgoing sessions: 0
- Host Details
 - Queried DNS names : 131.75.168.192.in-addr.arpa, 132.75.168.192.in-addr.arpa
 - Web Server Banner 1 : TCP 80 : Apache/2.2.8 (Ubuntu) DAV/2
 - FTP Server Banner 1 : TCP 21 : (vsFTPd 2.3.4)
 - Preferred SMB dialect : NT LM 0.12
 - SMB Native LAN Manager : Samba 3.0.20-Debian
 - SMB Native OS : nix
 - SSH Application : OpenSSH_4.7p1 Debian-8ubuntu1
 - SSH Version : 2.0

192.168.75.255

202.11.16.167 [jprs.jp]

202.222.203.169 [www.hitachi-systems.com]

Case Panel

Filename	MD5
network...	60e5a6b...

[Reload Case Files]

Live Sniffing Buffer Usage: [Progress Bar]

ファイル抽出

NetworkMiner 1.6.1

File Tools Help

--- Select a network adapter in the list --- Start Stop

Parameters (236) Keywords Cleartext Anomalies

Hosts (51) Frames (42xx) Files (18) Images (8) Messages Credentials (4) Sessions (1110) DNS (76)

Frame nr.	Reconst...	Source ...	S. port	Destina...	D. port	Protocol	Filename	Extensi...	
2726	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	robots.txt.html	html	2!
2730	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	HEAD.html	html	2!
2708	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	index.html	html	8!
2900	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	favicon.ico.html	html	2!
3052	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	index[1].html	html	8!
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	aus3.mozilla.or...	cer	1 2!
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	Thawte SSL C...	cer	1 1!
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	thawte Primar...	cer	1 0!
3854	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	index.html	html	24 6!
3881	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	styles.css	css	10 1!
3910	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	ic_shuryo_s.gif	gif	2!
3912	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	ic_new_s.gif	gif	2!
3938	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	kddi.png	png	6 8!
3952	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	dwango.png	png	4 9!
3970	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	isc2.png	png	7 9!
3986	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	jinsoken.png	png	6 0!
4005	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	temptech.png	png	7 2!
4015	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	jipdec.png	png	3 0!

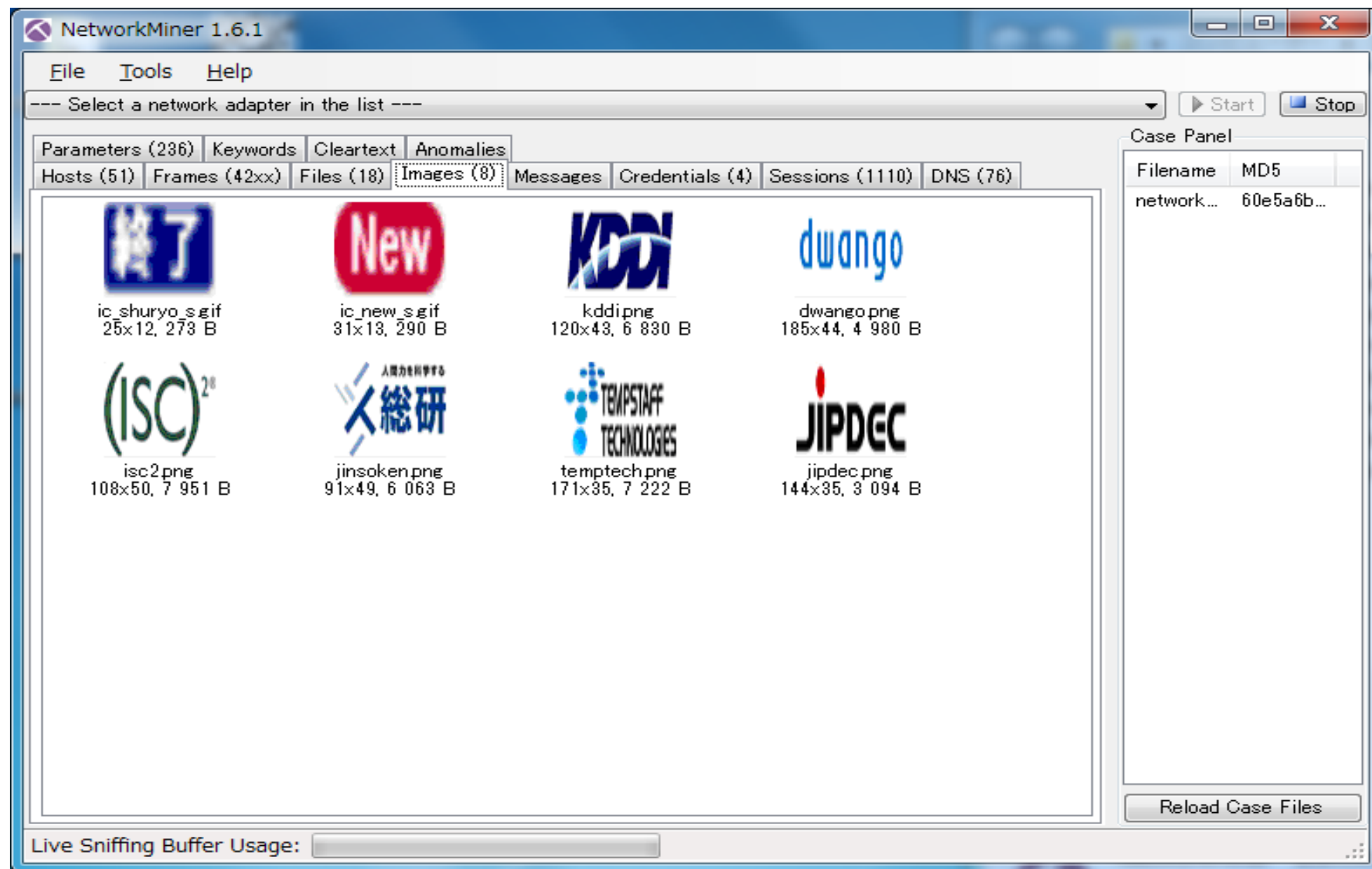
Case Panel

Filename	MD5
network...	60e5a6b...

Reload Case Files

Live Sniffing Buffer Usage:

画像



認証情報

The screenshot shows the NetworkMiner 1.6.1 application window. The 'Credentials (4)' tab is selected, displaying a table of captured login information. The table has columns for Client, Server, Protocol, Username, Password, Valid lo..., and Login ti....

Client	Server	Protocol	Username	Password	Valid lo...	Login ti...
192.168.7...	192.168...	FTP	anonymous	IEUser@	Unknown	2014/10...
192.168.7...	192.168...	IRC	(IRC User: nm...	N/A	Unknown	2014/10...
192.168.7...	192.168...	IRC	swighqprp(IRC...	N/A	Unknown	2014/10...
192.168.7...	192.168...	CIFS Setup An...	guest	AD2FE0BB...	Unknown	2014/10...

On the right, the 'Case Panel' shows a table with 'Filename' and 'MD5' columns, containing one entry: 'network...' with MD5 '60e5a6b...'. Below this is a 'Reload Case Files' button.

At the bottom, a 'Live Sniffing Buffer Usage' progress bar is visible.

Network Miner

メリット・デメリット

- メリット
 - ネットワークの知識が少なくても扱いやすい
 - ホストの情報を簡単にパケットから表示
 - 画像のサムネイルが表示できる
- デメリット
 - Windowsでしか使えない（工夫すれば他のOSでも使えるが…）
 - フィルタリング機能がない
 - ネットワークの勉強には不向き
 - pcap-ng形式のファイルは扱えない(有償版なら可)

WiresharkとNetwork Minerをうまく併用

付録 2

ネットワーク問題に使えるツール

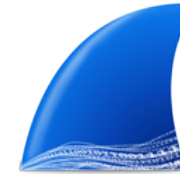
パケットキャプチャツール

- tcpdump (Linux, Mac OS XなどのUNIX系OS)
 - CUIでパケットキャプチャ、pcapファイルを簡易表示することも
 - 軽い
 - <http://www.tcpdump.org/>
- Windump (Windows)
 - tcpdumpのWindows移植版
 - <http://www.winpcap.org/windump/>
- dumpcap / tshark / Wireshark
 - Wiresharkをダウンロードすると付属
 - dumpcapとtsharkはCUI, WiresharkはGUI
 - 機能的には、dumpcap < tshark < Wireshark

パケット解析ツール

- Wireshark

- GUIでパケットキャプチャ、豊富なパケット解析機能
- tcpdumpと比べると重い
- <https://www.wireshark.org/>



- Network Miner

- パケットキャプチャ、ファイル抽出機能など
- Wiresharkに比べると機能が少ない
- <http://www.netresec.com/?page=NetworkMiner>



- Scapy / dpkt

- Pythonのモジュール
- Pcapファイルをパースして、Pythonで解析することが可能
- scapy : <http://www.secdev.org/projects/scapy/>
- dpkt : <https://code.google.com/p/dpkt/>

パケット生成・送信ツール

- nmap
 - ネットワークスキャンツール
 - <http://nmap.org/>
- hping
 - Pingのようなインターフェースでパケットを生成できる
 - <http://www.hping.org/>
- Scapy / dpkt
 - パケット関連のPythonライブラリ
 - パケットをパースするのみでなく、パケットの中身を操作して送信できる
- netcat (nc)
 - ネットワークを扱う万能ツール
 - 様々な種類のnetcatが存在する
 - 参考 : <http://d.hatena.ne.jp/EijiYoshida/20111109/1320800716>

Wireshark付属のツール

- editcap
 - pcapngファイルからpcapファイルへの変換
 - 参考 : <http://divisionbyzero.hatenablog.jp/entry/2012/09/03/223000>
 - pcap, pcapngファイルの分割
- mergcap
 - pcap, pcapngファイルの結合
 - 参考 : <http://d.hatena.ne.jp/giugno/20110914/1315983399>
- text2pcap
 - テキスト形式のパケットをpcap形式に変換

その他ツール

- pcapfix
 - 破損しているpcapファイルを修復
 - <https://f00l.de/pcapfix/>
- tcpreplay
 - pcapファイルに保存されているパケットを再送可能
 - pcapファイルのパケットの情報を書き換えることができる機能
 - <http://tcpreplay.jp/>