

# CTF for ビギナーズ

ネットワーク講習

---

# 自己紹介

---

- 保要 隆明 (ほよう たかあき)
- Hから始まる理工系大学院 修士課程2年
- CTF team : \*\*\*\*\* → ???
- twitter : @takahoyo
- CTF4b歴 : 2年目



# 本日の内容

---

1. ネットワーク分野で必要な知識と技術は？
  - パケット・通信プロトコル
  - Wiresharkの基本的な使い方
2. CTFにおけるネットワーク問題
  - どのような問題が出るのか
3. 問題を見てみよう
  - 問題解くときに見るべきポイント
4. 今後のレベルアップするためには

# 本日の内容

---

1. ネットワーク分野で必要な知識と技術は？
  - パケット・通信プロトコル
  - Wiresharkの基本的な使い方
2. CTFにおけるネットワーク問題
  - どのような問題が出るのか
3. 問題を見てみよう
  - 問題解くときに見るべきポイント
4. 今後のレベルアップするためには

# ネットワークで必要な知識と技術

---

- ネットワークプロトコルについての知識
- ネットワークを流れる通信を解析する技術

# 通信プロトコル

---

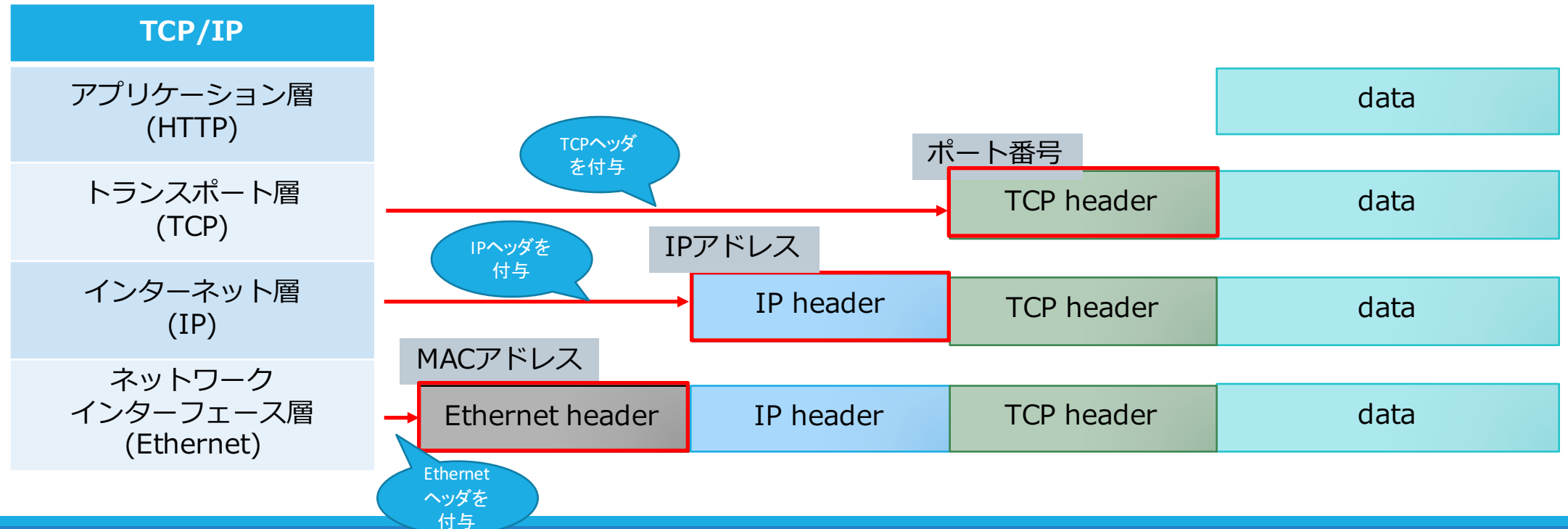
# 通信プロトコル

- 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

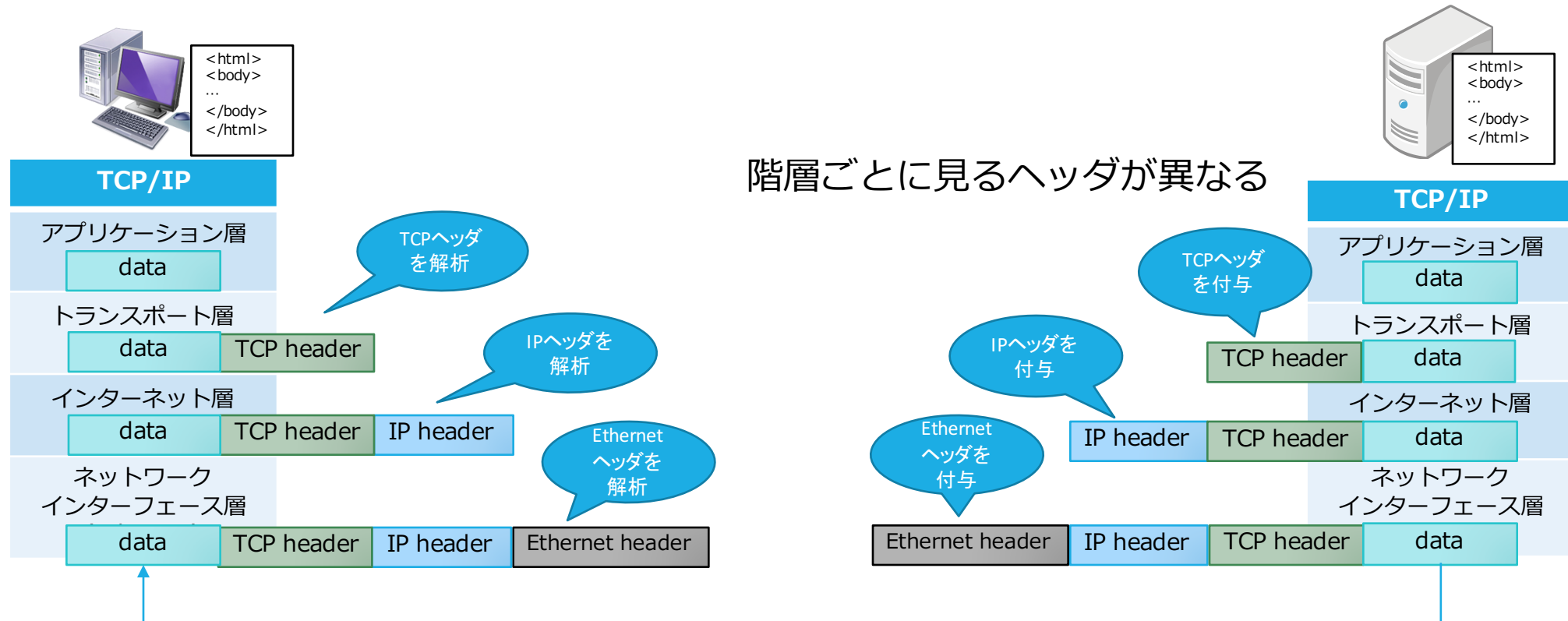
- 各プロトコルの階層ごとに制御情報(Header)が付与
- Ex. Webページ閲覧





# TCP/IP通信の例

- Ex. Webページの閲覧

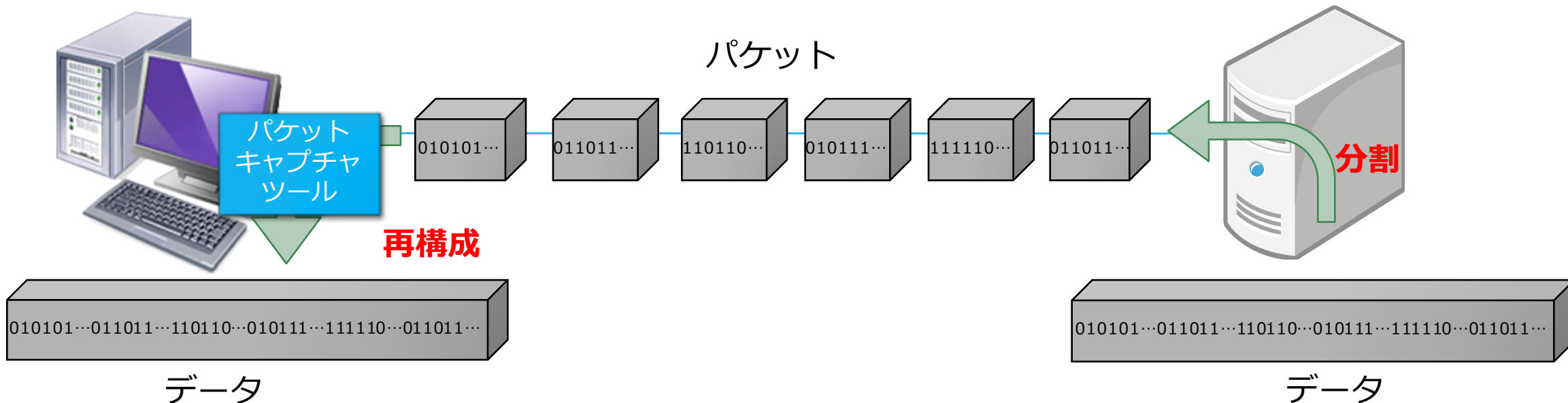


# パケット

---

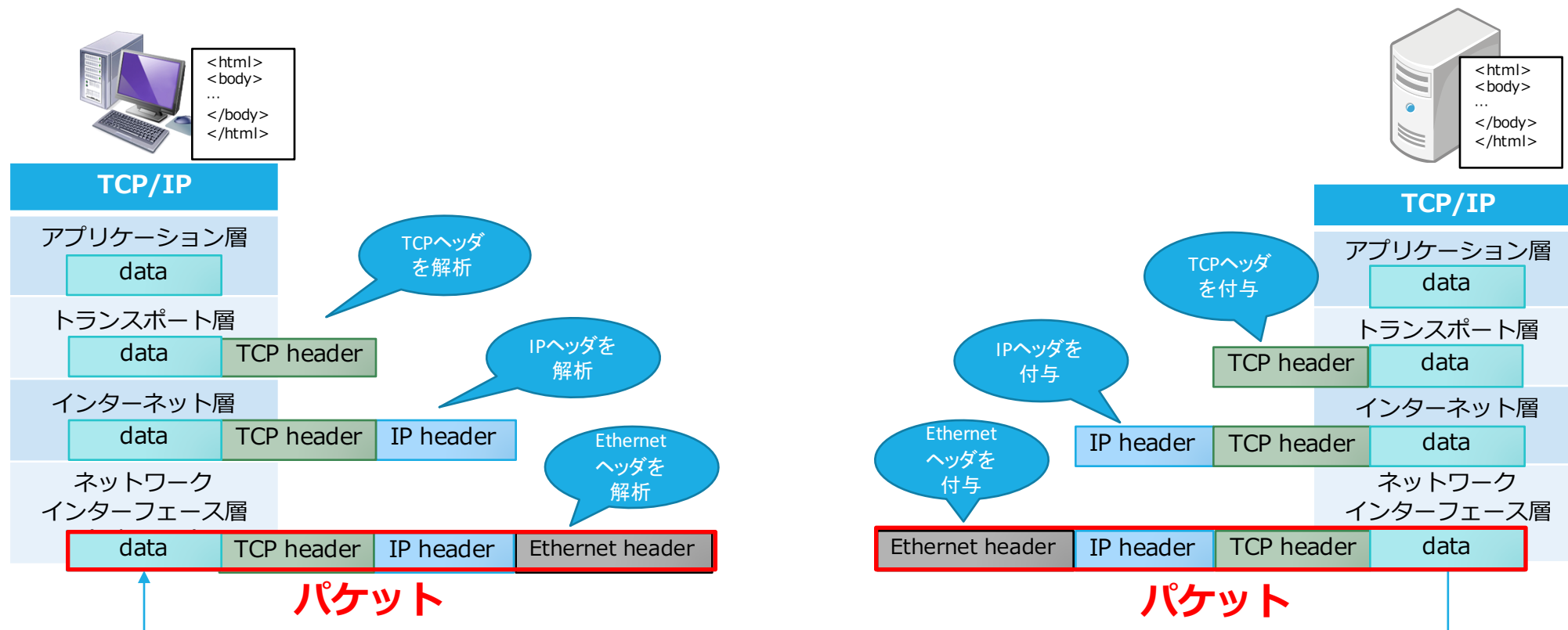
# パケット(packet)とは

- 直訳 → 小包、小箱の意味
- ネットワーク上に分割されたデータ = **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



# 先ほどの例で言うと...

- Ex. Webページの閲覧



# 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
- CTFでは、このファイル形式で出題されることが多い  
→ これを読めるようにならないといけない
- 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)
```

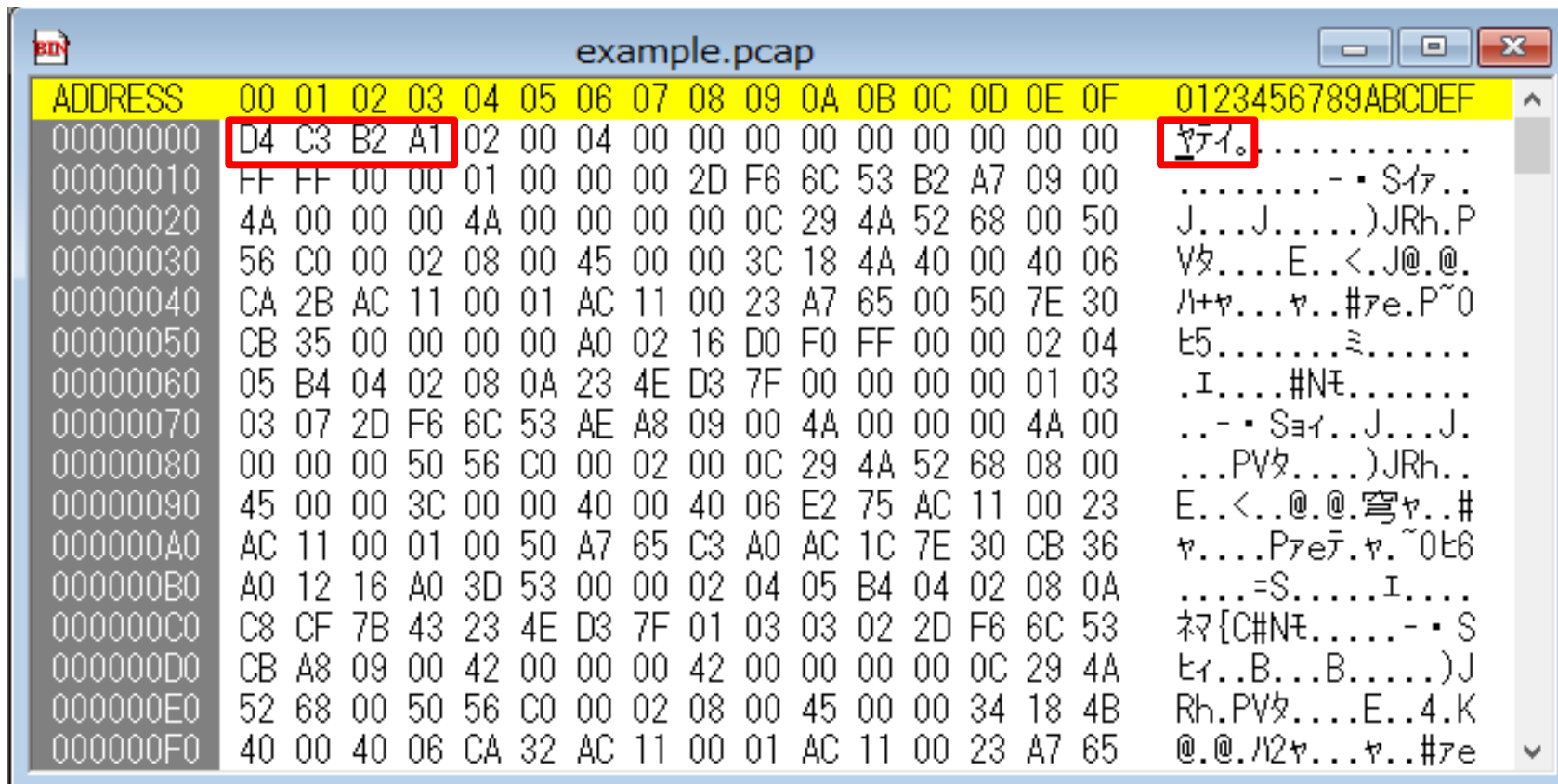
```
$ file example.pcapng
example.pcapng: pcap-ng capture file - version 1.0
```

※fileコマンド：ファイルの種類をデータの内容から判定するコマンド

※ Macのデフォルトの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？

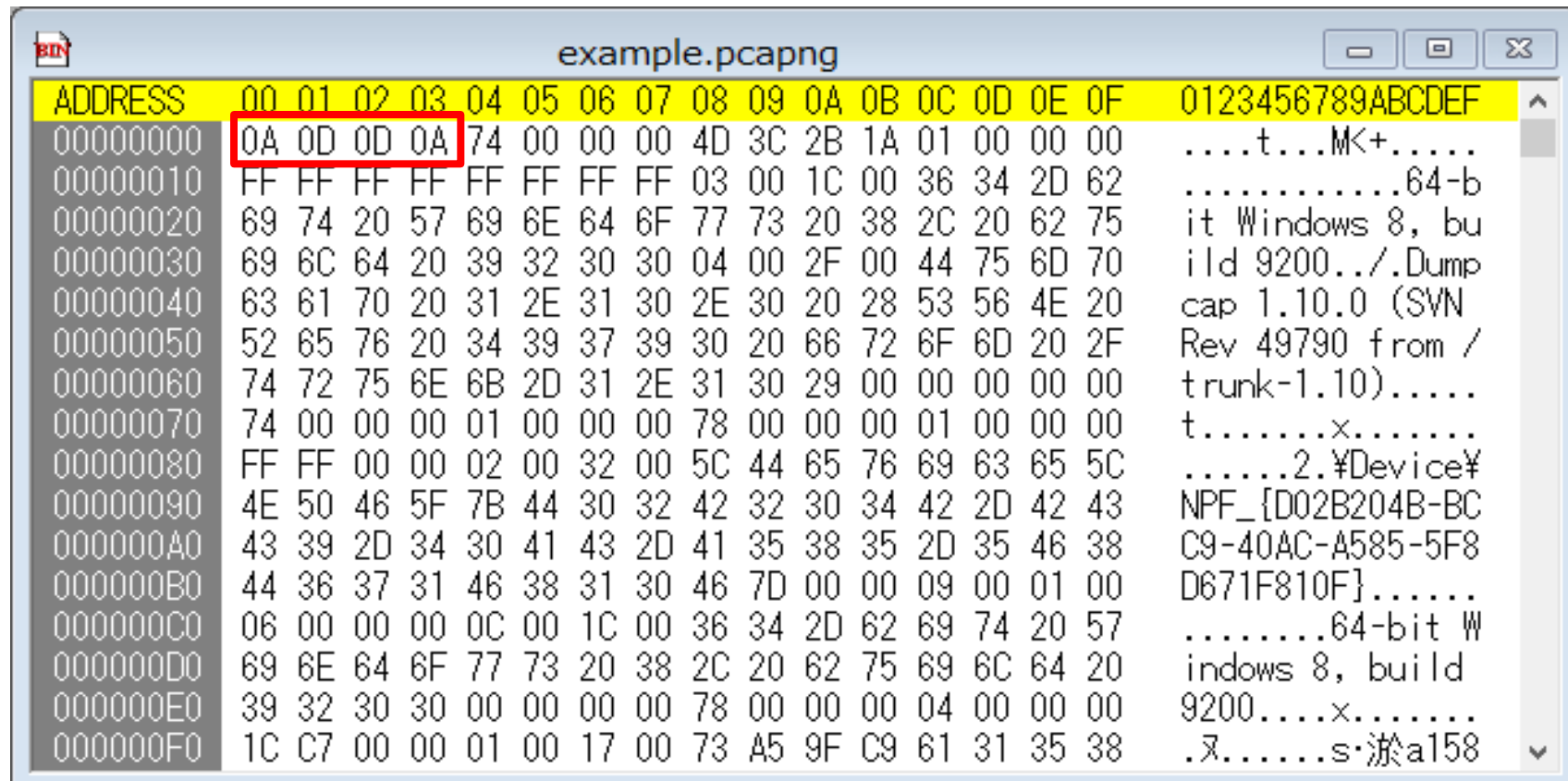
- pcapファイルをバイナリエディタで見ると…



※バイナリエディタ  
Strilingを使っています

# pcap(pcap-ng)ファイル？

- pcap-ngファイルをバイナリエディタで見ると…



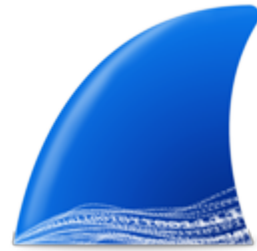
※バイナリエディタ  
Strilingを使っています

# pcapファイルを読むためには…

---

- 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
- ~~バイナリエディタを使用する（上級者向け）~~





# Wireshark

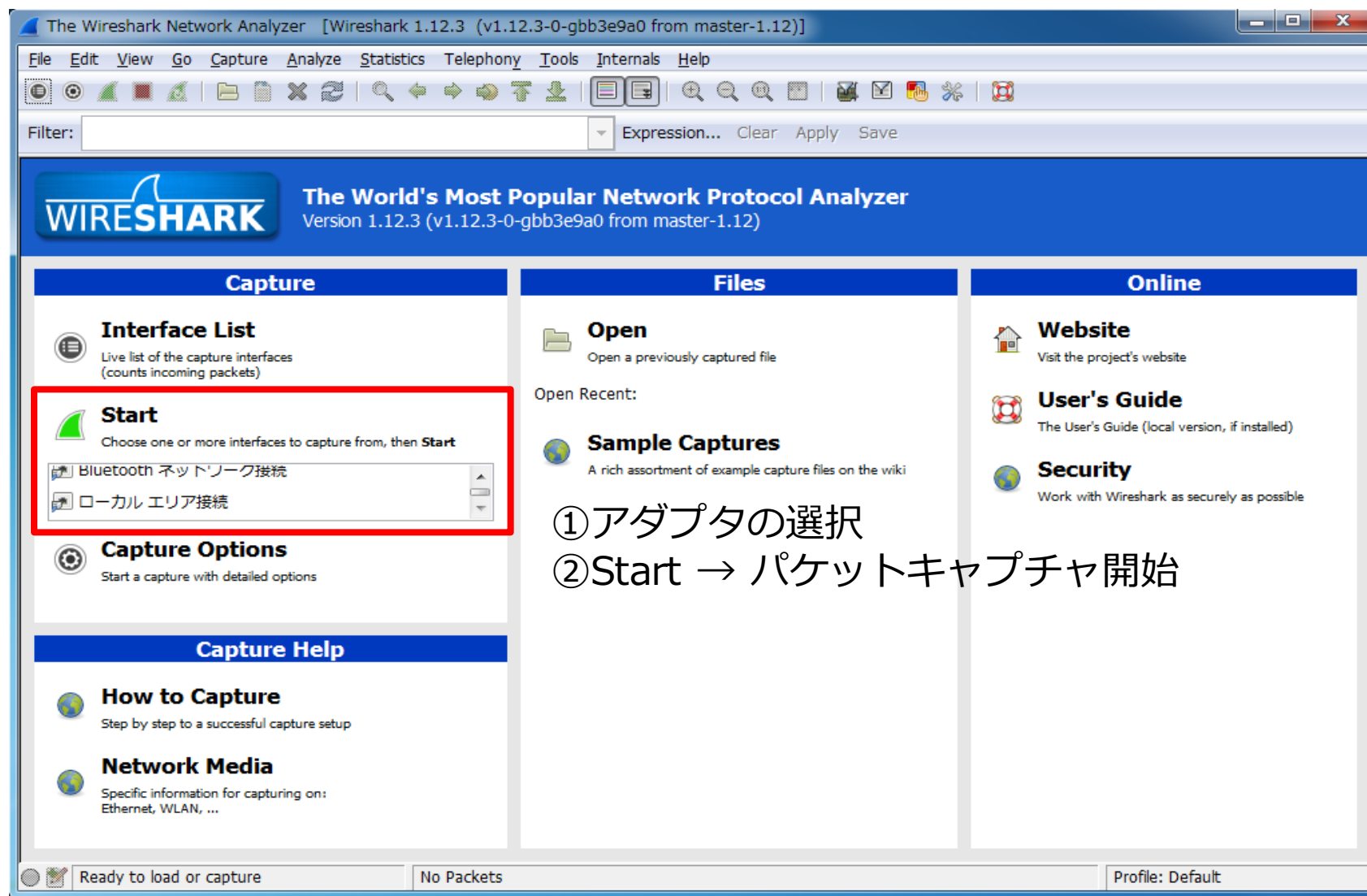
---

# 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



# 詳細なパケット表示機能

Display Filter

ディスプレイフィルタ

Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)

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.seccon.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 63 .....2 014.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

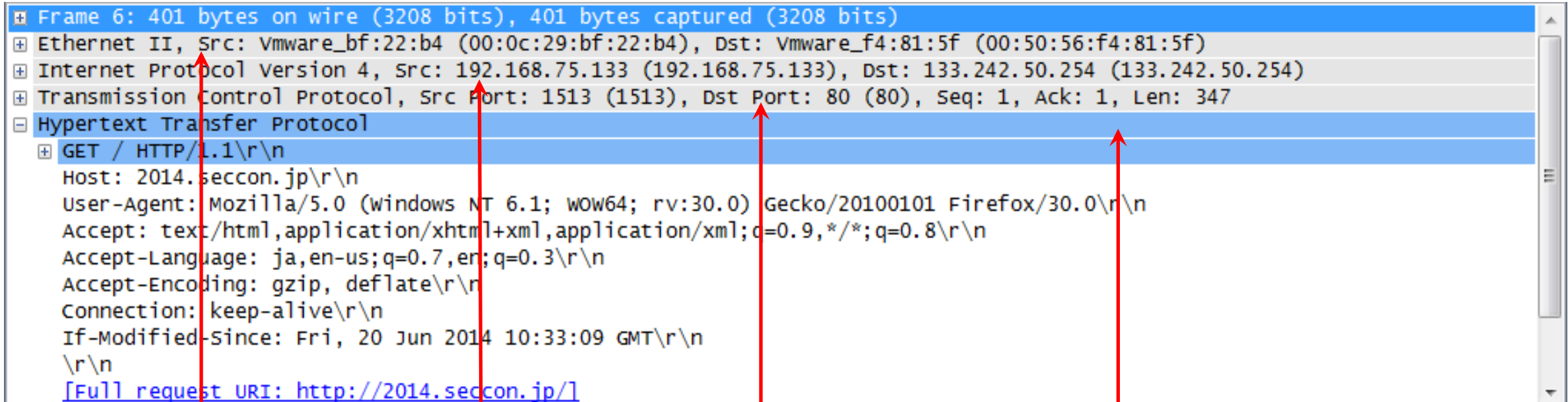
1513+80 [SYN]

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=14
4	0.050884	133.242.50.254	192.168.75.133	TCP	60	http-fujitsu-dtc [SYN, ACK] Seq=1
5	0.050961	192.168.75.133	133.242.50.254	TCP	54	fujitsu-dtc-http [ACK] Seq=1 Ack=348 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	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)



パケットの各ヘッダ情報  
と対応



# Packet Bytes (生のパケット)

- Wiresharkが解析し、整形してくれている
- 場合によっては見る必要アリ

0000	00	50	56	f4	81	5f	00	0c	29	bf	22	b4	08	00	45	00
0010	01	23	77	75	40	00	80	06	00	00	c0	a8	4b	85	85	f2
0020	32	fe	05	19	00	50	ce	68	b9	57	2e	39	4e	76	50	18
0030	fa	f0	c6	33	00	00	47	45	54	20	2f	20	48	54	54	50
0040	2f	31	2e	31	0d	0a	41	63	63	65	70	74	3a	20	74	65
0050	78	74	2f	68	74	6d	6c	2c	20	61	70	70	6c	69	63	61
0060	74	69	6f	6e	2f	78	68	74	6d	6c	2b	78	6d	6c	2c	20
0070	2a	2f	2a	0d	0a	41	63	63	65	70	74	2d	4c	61	6e	67
0080	75	61	67	65	3a	20	6a	61	2d	4a	50	0d	0a	55	73	65
0090	72	2d	41	67	65	6e	74	3a	20	4d	6f	7a	69	6c	6c	61
00a0	2f	35	2e	30	20	28	63	6f	6d	70	61	74	69	62	6c	65
00b0	3b	20	4d	53	49	45	20	39	2e	30	3b	20	57	69	6e	64
00c0	6f	77	73	20	4e	54	20	36	2e	31	3b	20	57	4f	57	36
00d0	34	3b	20	54	72	69	64	65	6e	74	2f	35	2e	30	29	0d
00e0	0a	41	63	63	65	70	74	2d	45	6e	63	6f	64	69	6e	67
00f0	3a	20	67	7a	69	70	2c	20	64	65	66	6c	61	74	65	0d
0100	0a	48	6f	73	74	3a	20	32	30	31	34	2e	73	65	63	63

```
.PV..._... ). "...E.  
.#wu@... ....K...  
2....P.h .W.9NvP.  
...3..GE T / HTTP  
/1.1..Ac cept: te  
xt/html, applica  
tion/xht ml+xml,  
*/*..Acc ept-Lang  
uage: ja -JP..Use  
r-Agent: Mozilla  
/5.0 (co mpatible  
; MSIE 9 .0; wind  
ows NT 6 .1; WOW6  
4; Tride nt/5.0).  
.Accept- Encoding  
: gzip, deflate.  
.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で送受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



# 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

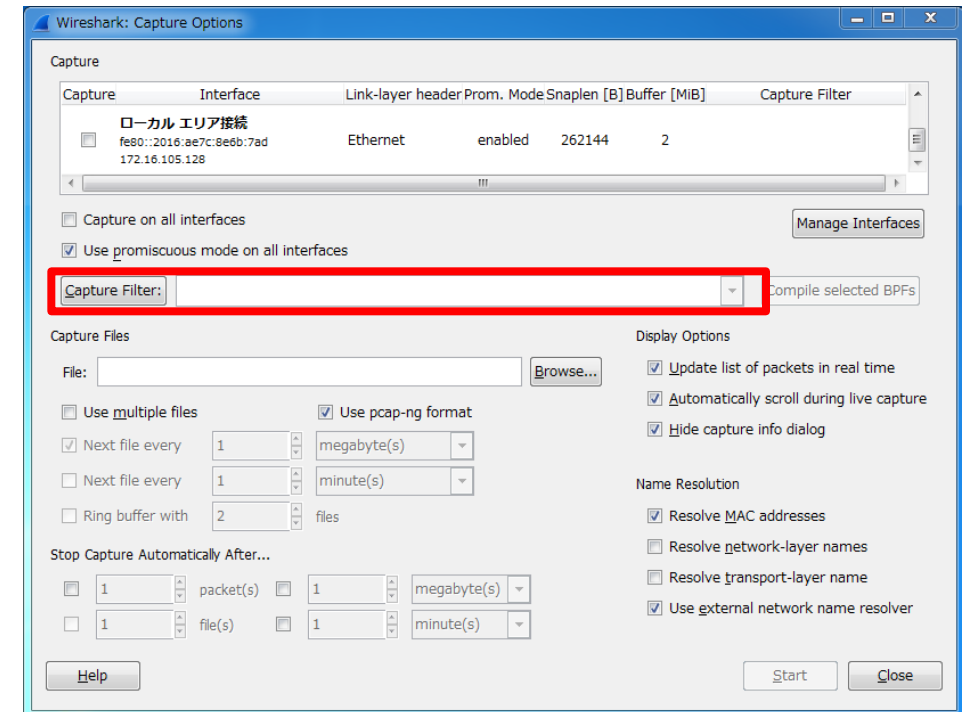
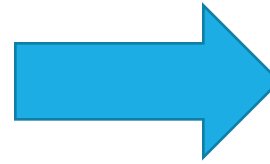
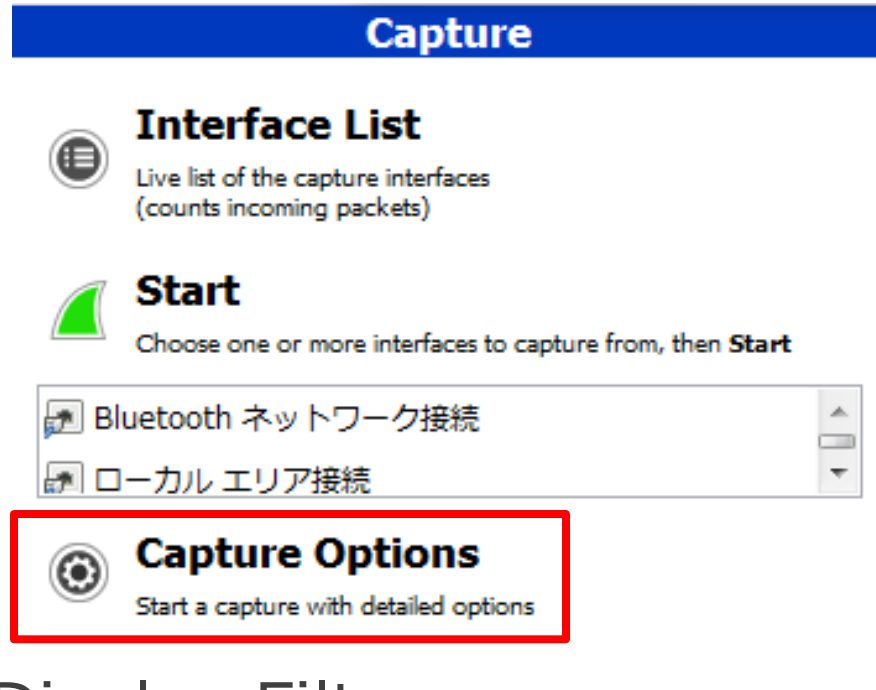
Filter: tcp

Filter: udp

Filter: http

# 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
  - 今回は説明省略
- 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.response	HTTPのレスポンス

比較演算子

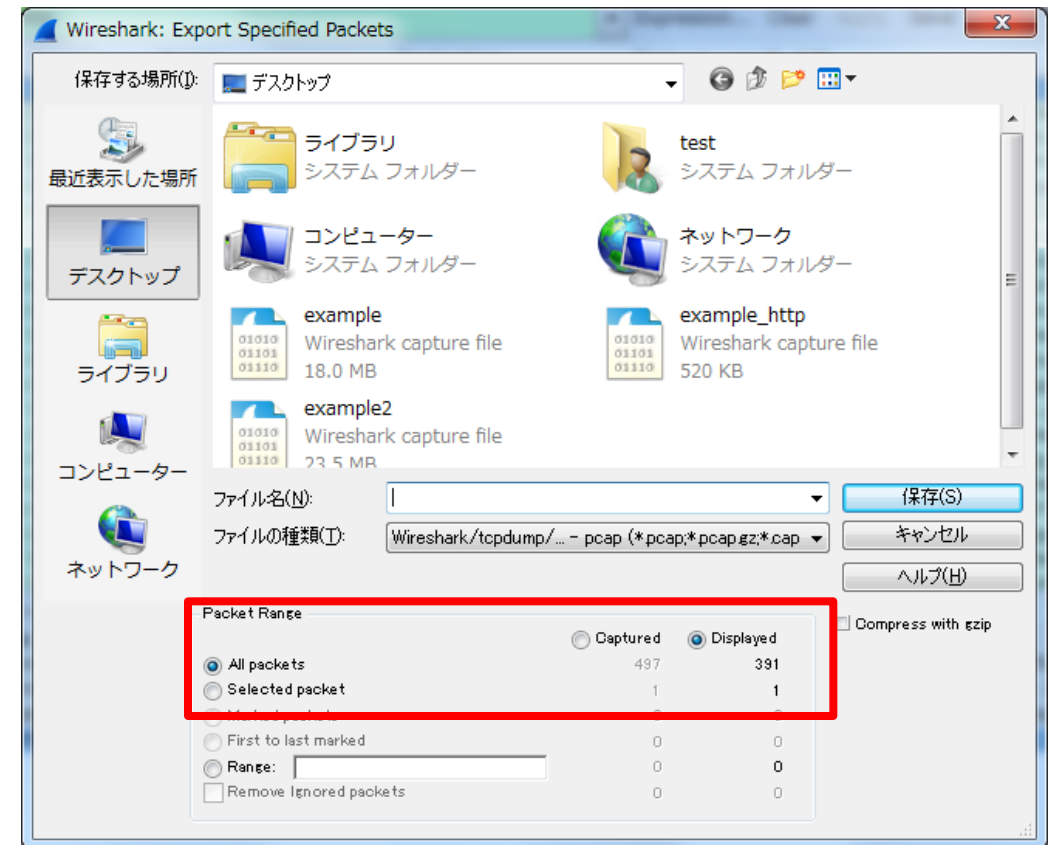
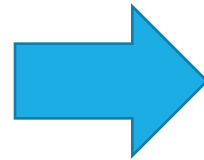
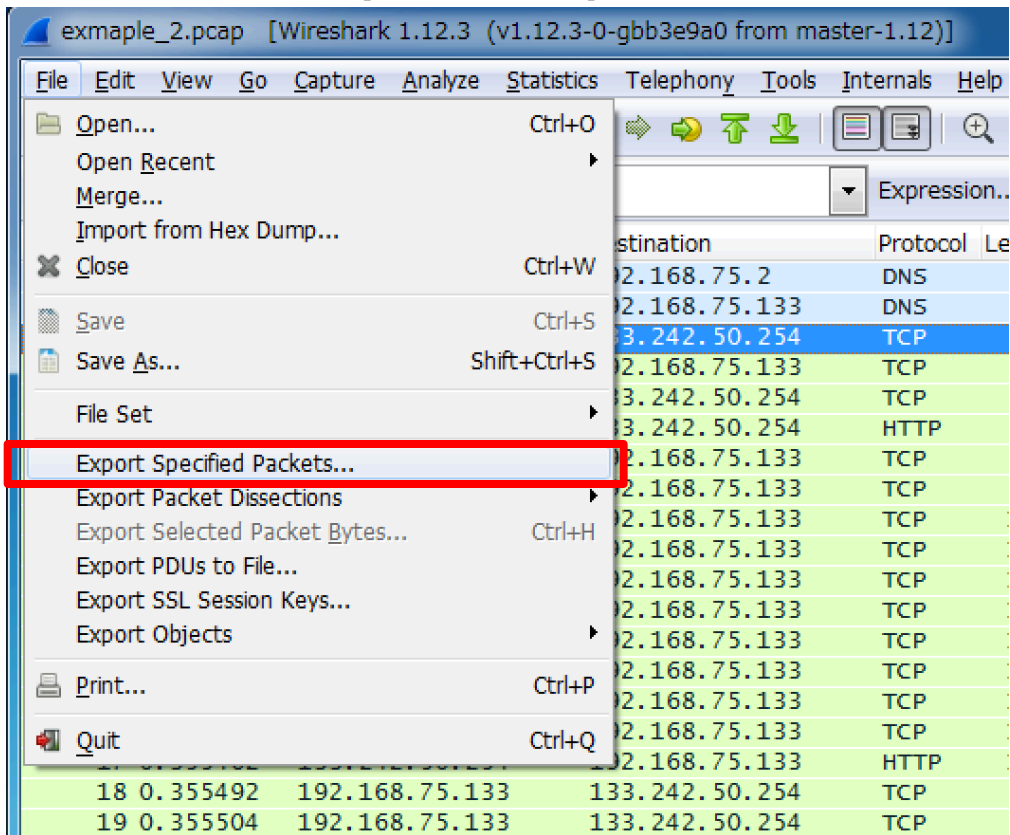
比較演算子	意味
eq (==)	等しい
ne (!=)	等しくない
gt (>)	大きい
lt (<)	小さい
ge (>=)	以上
le (<=)	以下

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

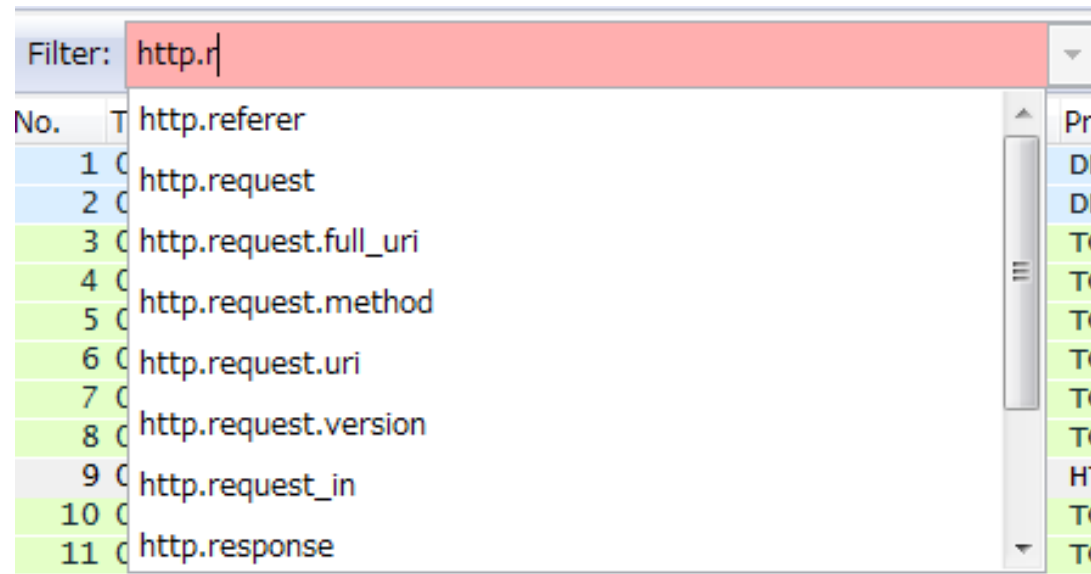
# Display Filter

- 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



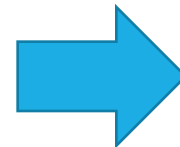
# TCPで送受信されるデータを抜きたい

- TCPパケットを選択して右クリック -> 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)

Mark Packet (toggle)  
Ignore Packet (toggle)  
Set Time Reference (toggle)  
Time Shift...  
Edit Packet  
Packet Comment...  
Manually Resolve Address  
Apply as Filter  
Prepare a Filter  
Conversation Filter  
Colorize Conversation  
SCTP  
**Follow TCP Stream**  
Follow UDP Stream  
Follow SSL Stream  
Copy  
Protocol Preferences  
Decode As...  
Print...  
Show Packet in New Window



Follow TCP Stream

Stream Content

```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: ja-JP
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Accept-Encoding: gzip, deflate
Host: 2014.secon.jp
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Sun, 22 Jun 2014 13:16:43 GMT
Content-Type: text/html
Last-Modified: Fri, 20 Jun 2014 10:33:09 GMT
Connection: keep-alive
Content-Length: 14384
Pragma: no-cache
Cache-Control: no-store, no-cache, must-revalidate, max-age=0
Expires: Tue, 01 Jan 1971 02:00:00 GMT

<!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">
<meta http-equiv="Cache-Control" content="no-cache">
<meta http-equiv="Expires" content="Thu, 01 Dec 1994 16:00:00 GMT">
<meta name="generator" content="Movable Type 5.2.10">
<link rel="stylesheet" href="/2014/styles.css" type="text/css">
<link rel="alternate" type="application/atom+xml" title="Recent Entries" href="/
```

Entire conversation (97205 bytes)

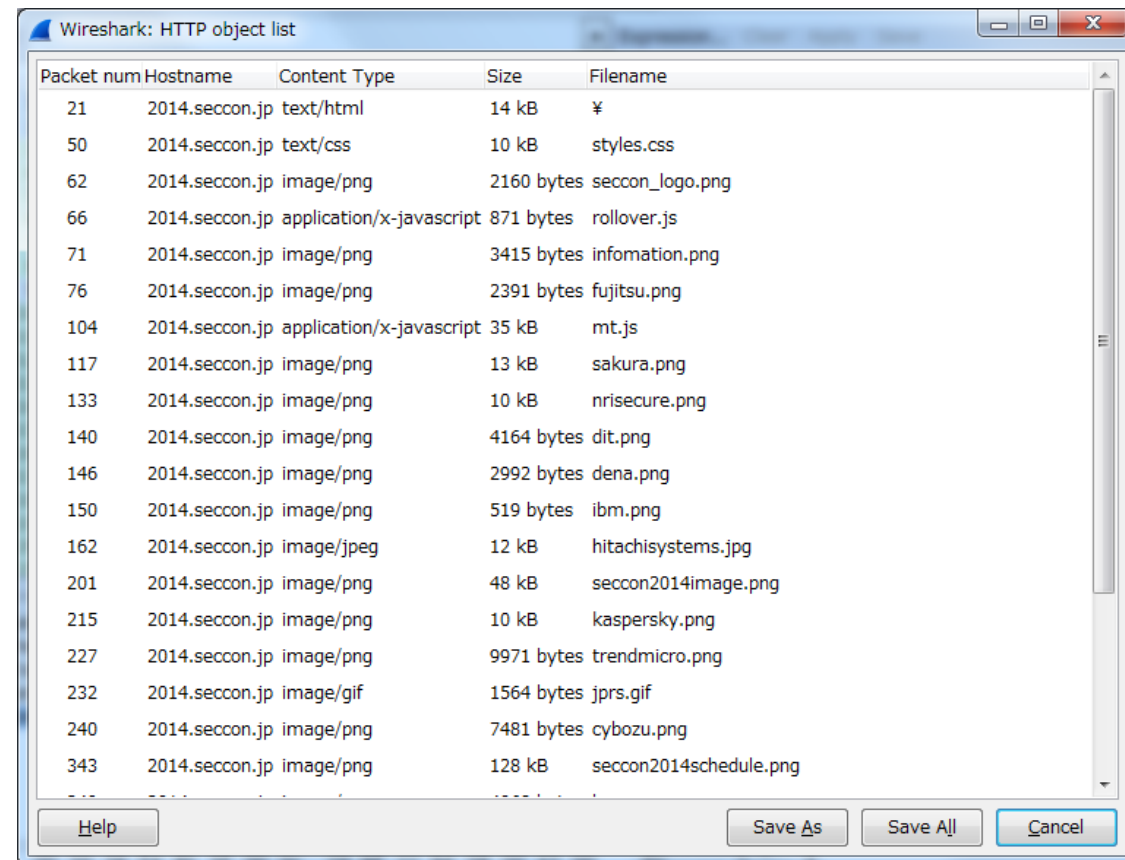
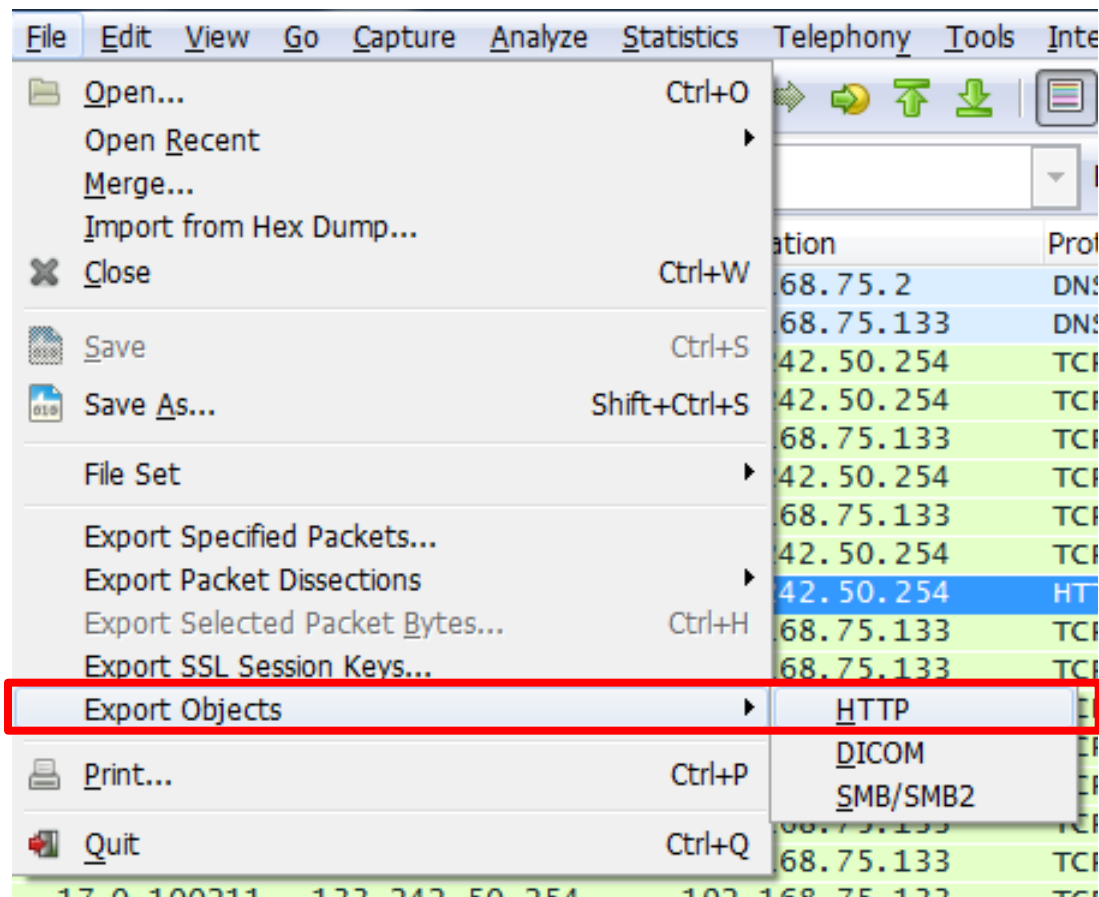
Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close



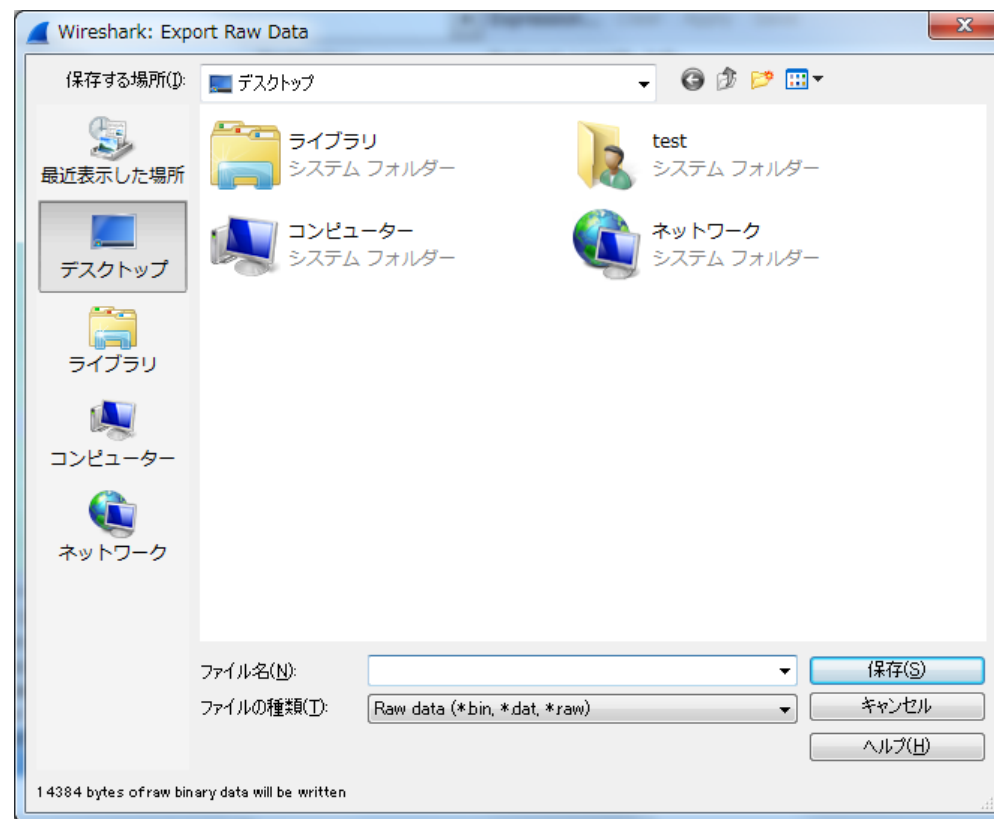
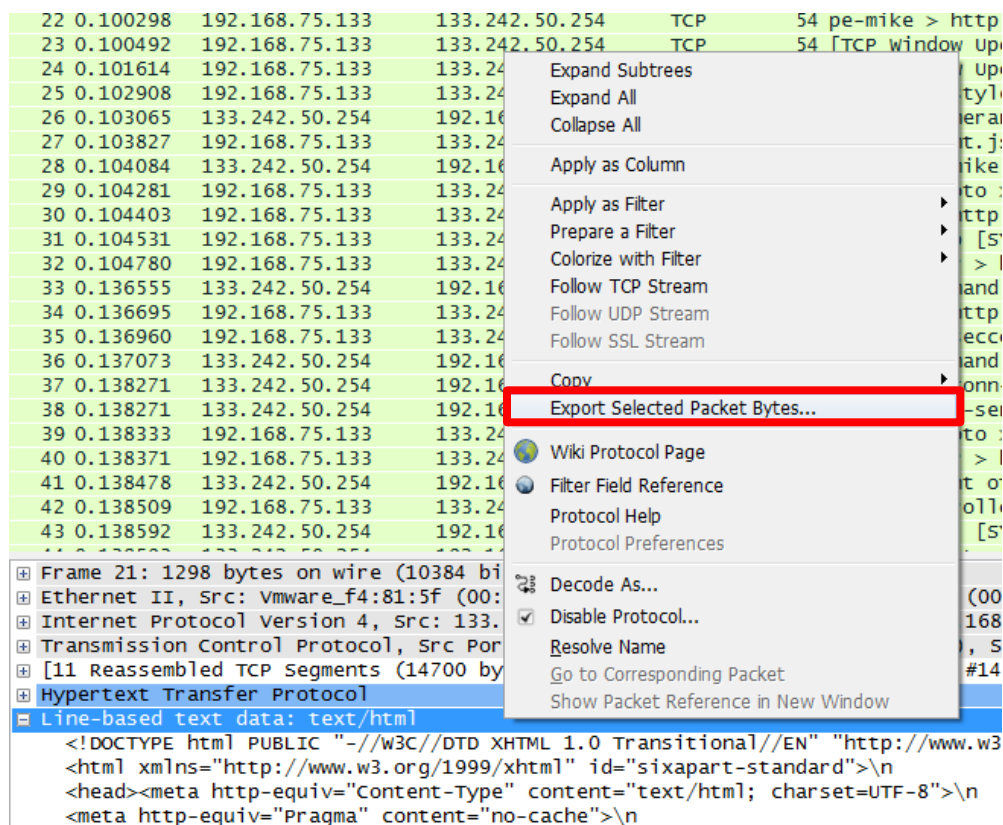
# HTTPで取り扱ってるファイルを抽出

- File -> Export Objects -> HTTP



# パケットから生データを抽出

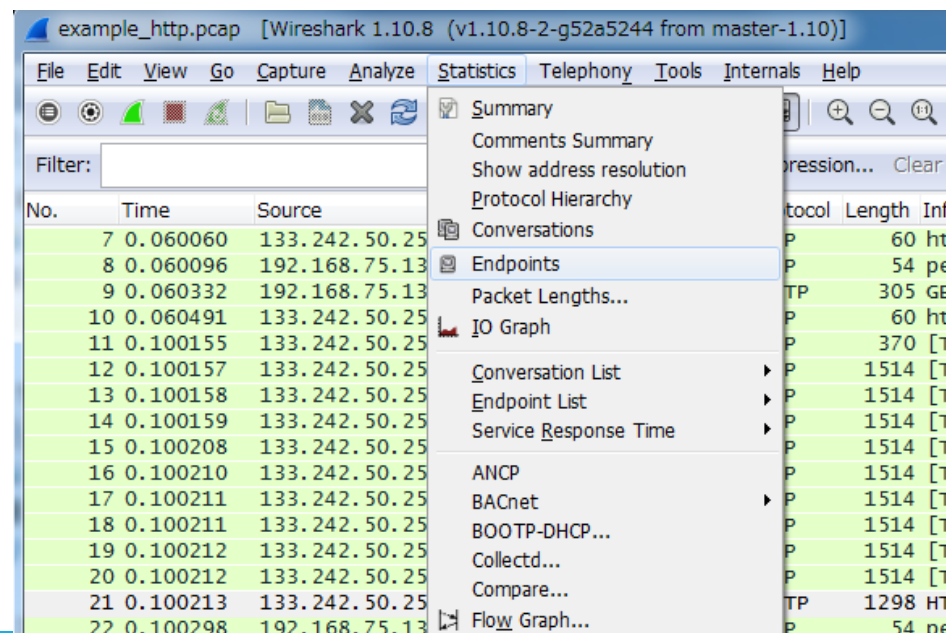
- パケットの詳細画面で、抽出したい部分を右クリック  
->Export Selected Packet Bytes



# Wiresharkの統計機能

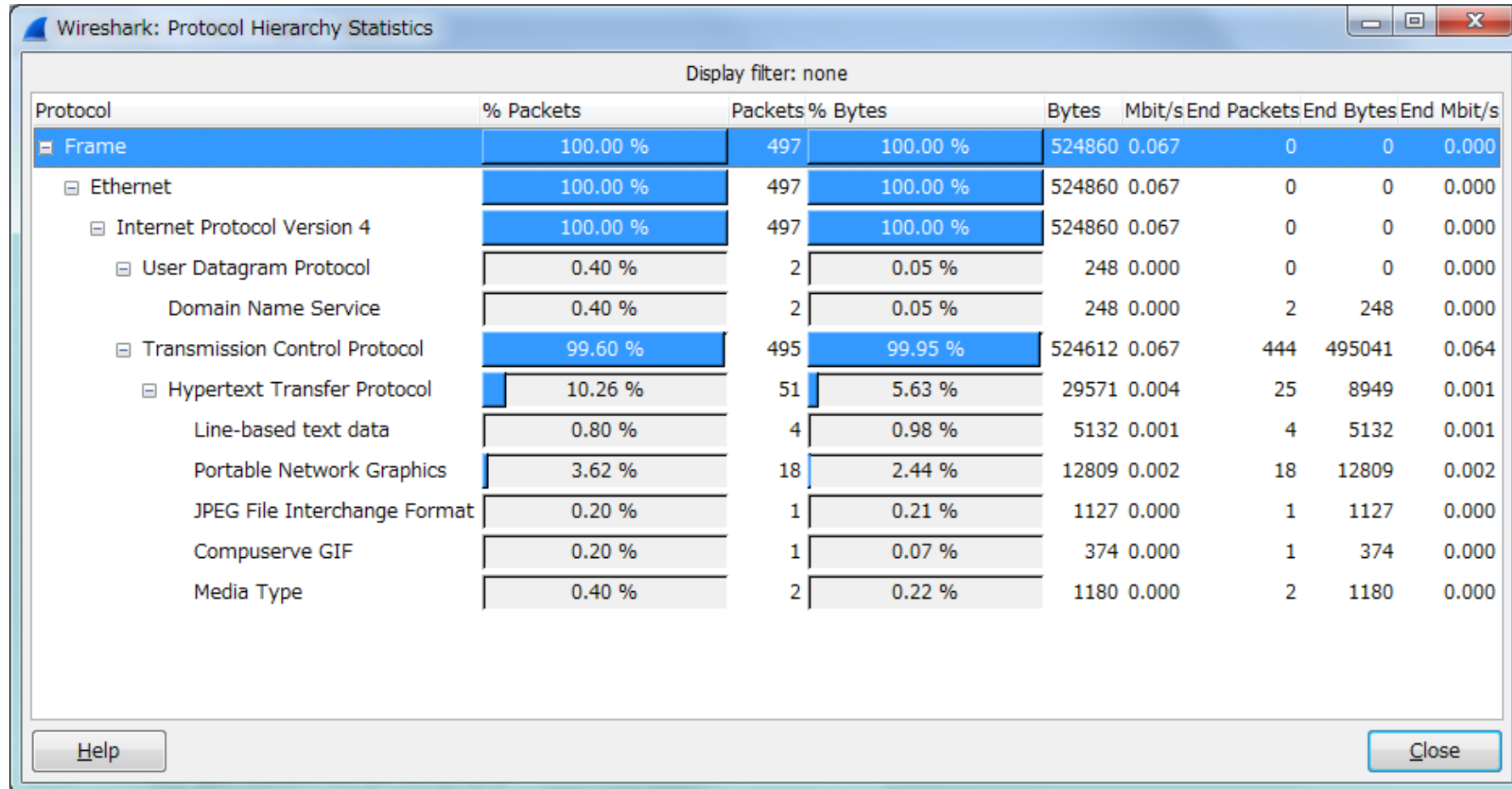
- Statistics

- 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



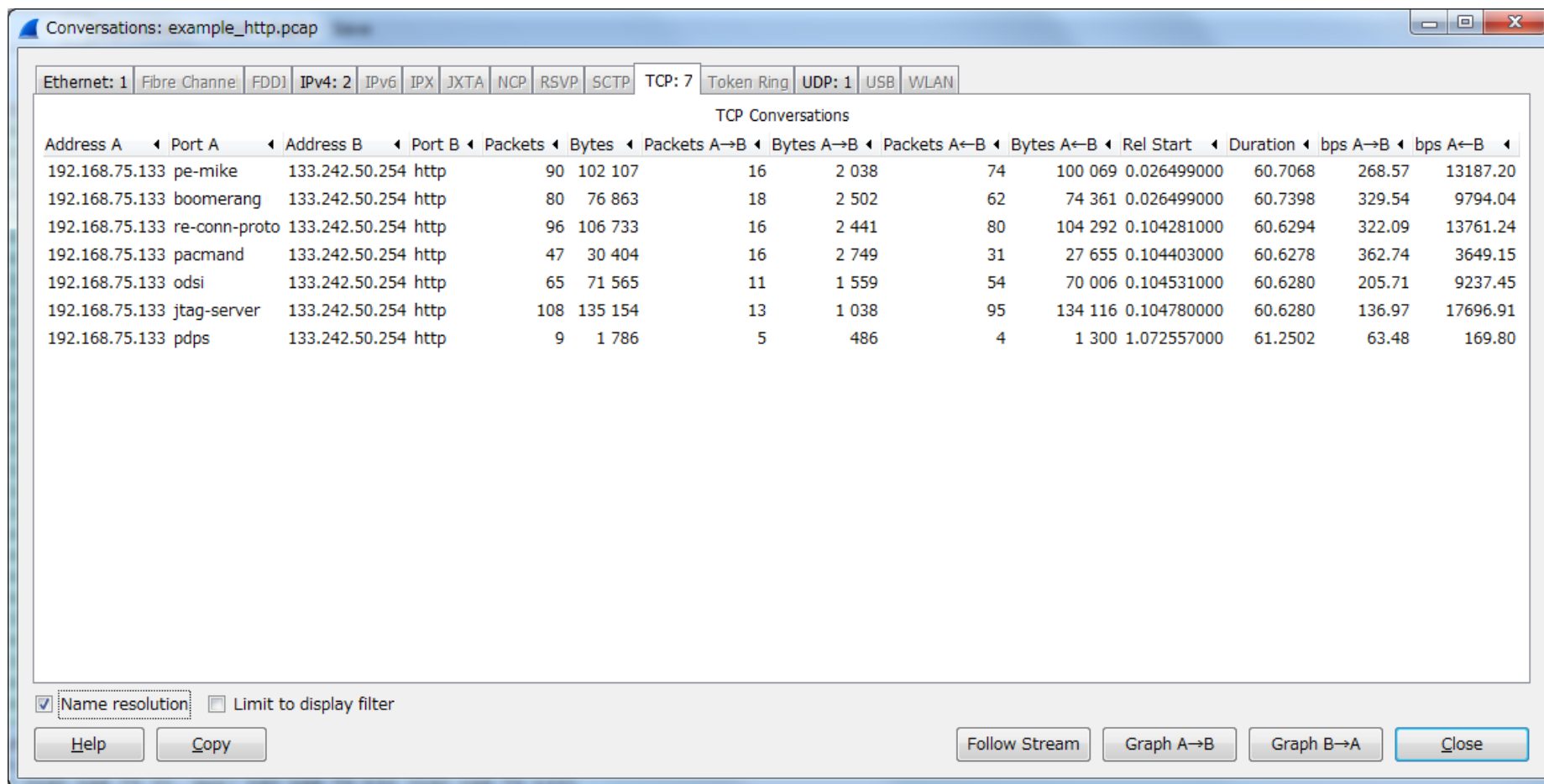
# 利用されているプロトコルの統計を知りたい

- Statistics -> Protocol Hierarchy



# 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

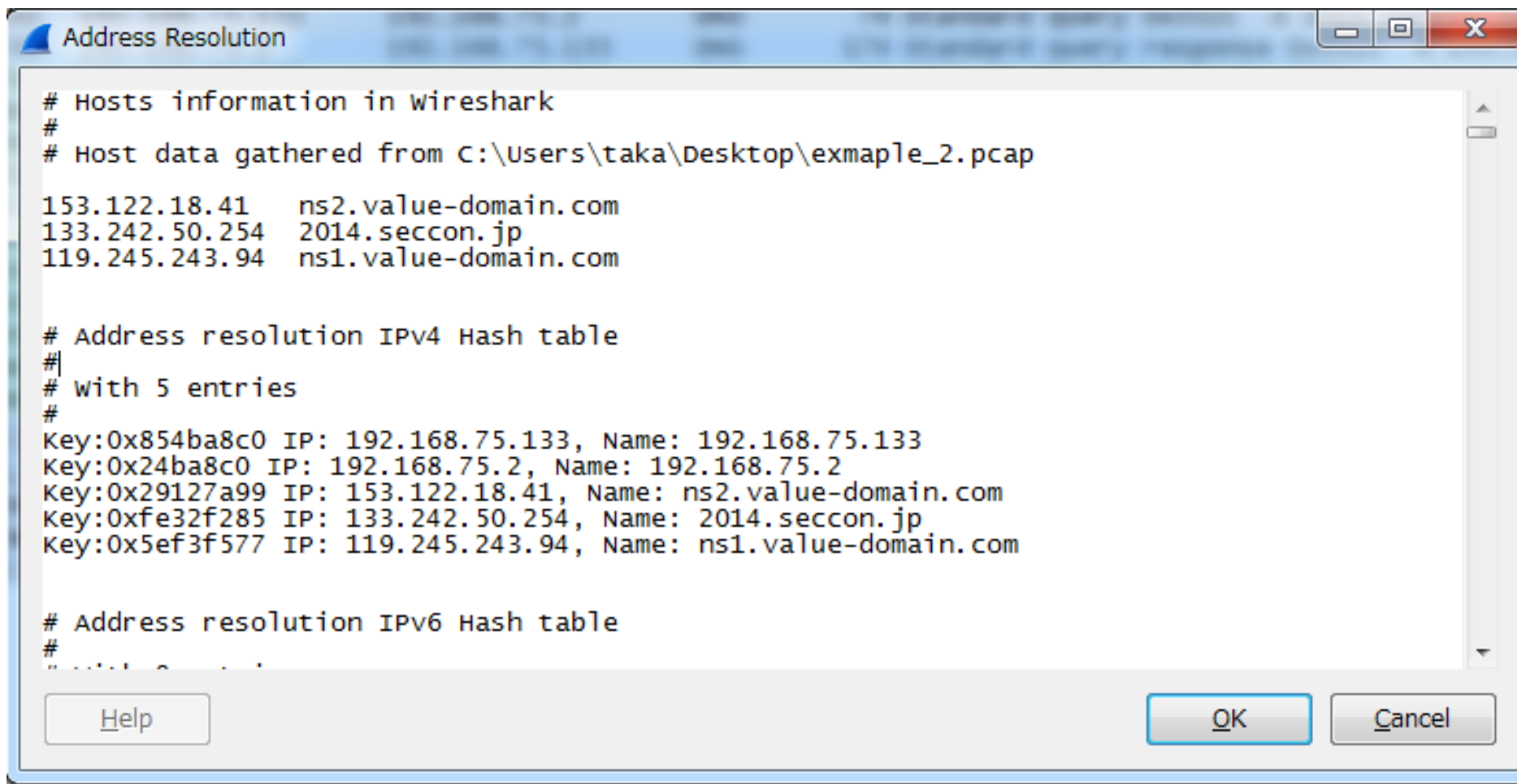


The screenshot shows the 'Conversations' window in Wireshark, filtered for 'example\_http.pcap'. The 'TCP' tab is selected, showing a list of conversations. The table below represents the data shown in the window.

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

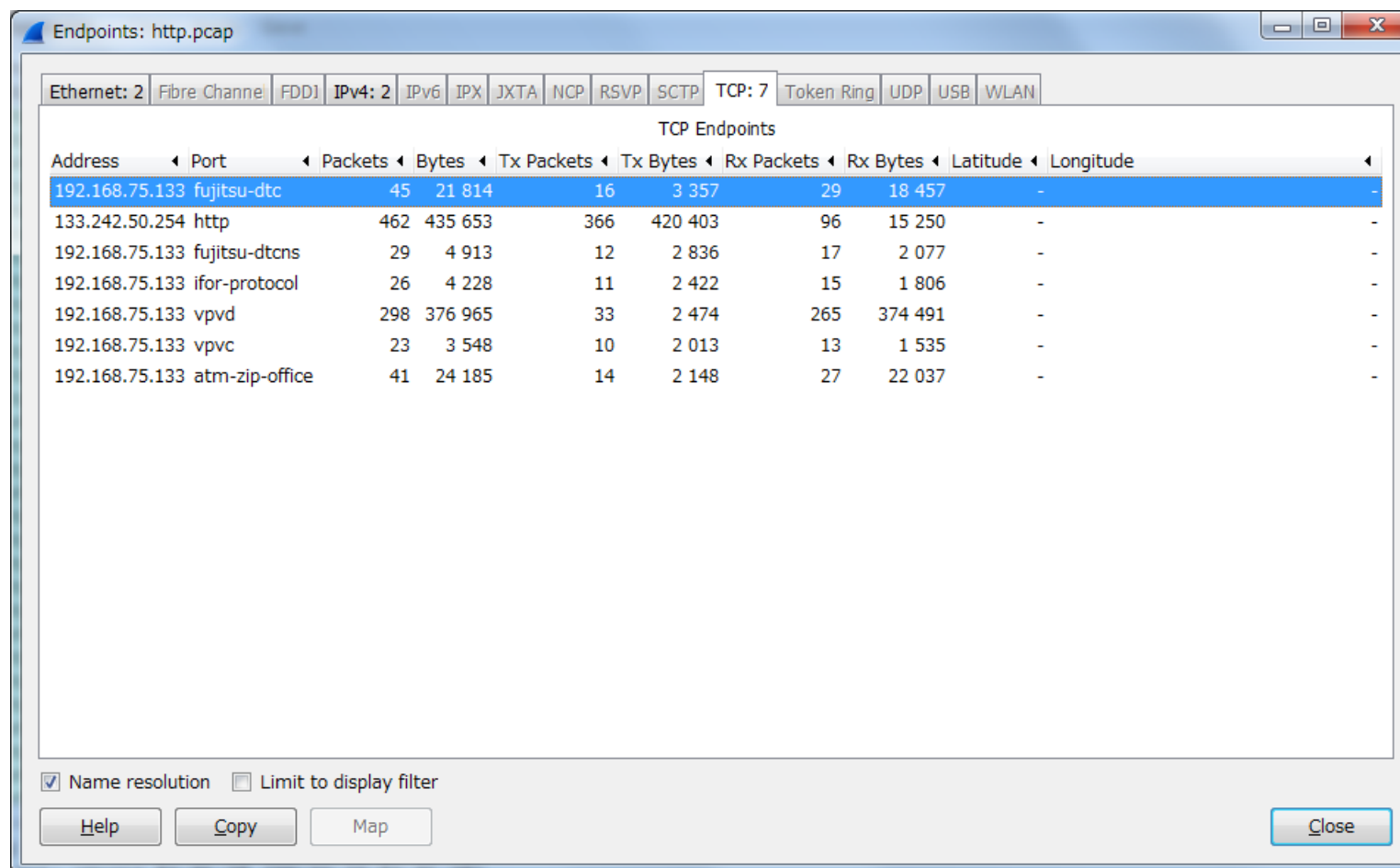
# IPアドレスとドメイン名を知りたい

- Statistics -> Show address resolution



# 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

# 本日の内容

---

1. ネットワーク分野で必要な知識は？
  - パケット・通信プロトコル
  - Wiresharkの基本的な使い方
2. CTFにおけるネットワーク問題
  - どのような問題が出るのか
3. 問題を見てみよう
  - 問題解くときに見るべきポイント
4. 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】

---

- 以下のような問題がある
- pcap(pcapng)ファイルを渡される問題
- サーバへ接続する問題

# pcap(pcapng)ファイルの場合

---

- TCP/IP, USB, Bluetoothなど多様な通信が記録されている
- これらをパケット解析して、Flagを見つける
- Networkというジャンルは最近のCTFではあまりない
- Forensicsとしてパケットの問題が出題されることが多い
- CryptoやWebの複合問題として出題されることがもしばしば
  - Ex. Web通信がパケットに記録されている

# サーバへ接続する問題

---

- パケットが問題として与えられるパターン
  - パケットを解析して、アクセスするサーバを決める
  - パスワード等の情報がパケットに記録されている場合もある
- アドレス情報が与えられているパターン
  - 書かれている情報を基にサーバへ接続する

# 本日の内容

---

1. ネットワーク分野で必要な知識は？
  - パケット・通信プロトコル
  - Wiresharkの基本的な使い方
2. CTFにおけるネットワーク問題
  - どのような問題が出るのか
3. 問題を見てみよう
  - 問題解くときに見るべきポイント
4. 今後のレベルアップするためには

# 問題を見るときのポイント

---

- パケットが与えられた問題の場合 (この場合のみ紹介)
  - IPアドレス
  - 使われているプロトコル
- アドレス情報が与えられた問題の場合
  - パケットキャプチャして通信を調べる
  - パケットの中身进行操作してアクセス

# IPアドレス

---

- オンラインCTFの場合
  - グローバルIPの通信 → そこに接続する問題の可能性が高い
  - プライベートIPの通信 → パケットの中に答えがある可能性が高い
- オフラインCTFの場合
  - 競技用ネットワークから到達できるIPの通信  
→ そこに接続する問題の可能性が高い
  - 競技用ネットワークから到達できないIPの通信  
→ パケットの中に答えがある可能性が高い
- Statisticsの“Conversations”機能を使うとわかりやすい

# 使われているプロトコル

---

- 暗号化されていないプロトコルを探す
  - http
  - telnet
  - ftp
  - smtp
  - など
- 暗号化されている場合、パケットキャプチャしても情報がわからないため(暗号解読は、cryptoのジャンル)
- Statisticsの"Protocol Hierarchy"機能を使うとわかりやすい

# 演習

---



# 問題を解いてみよう

---

- network\_ex.pcap
- 過去のCTF for ビギナーズ CTFで出題された問題
- パケットが与えられた問題
- 競技ネットワークに接続したら、IPアドレス172.20.x.xが振られた

# IPアドレスを見る

The image shows a Wireshark 1.12.4 window displaying a network capture file named 'network100.pcap'. The main packet list on the left shows several ICMP Echo (ping) requests. The packet details pane on the right shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. A 'Conversations: network100.pcap' dialog box is open, showing a TCP conversation between two hosts. The hosts are 192.168.29.134 (Port 41185) and 192.168.29.129 (Port 7777). The dialog box includes a table of packets and bytes for the conversation.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start
192.168.29.134	41185	192.168.29.129	7777	54	3 627	28	1 903	26	1 724	16.7757070

競技ネットワークではない  
プライベートIPアドレス

# プロトコルを見る

network100.pcap [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Src port	Dest port	Protocol	Length	Info
34	2014-10-28 17:43:37.747449	192.168.29.129	192.168.29.134			ICMP	98	Echo (ping) re
35	2014-10-28 17:43:38.746390	192.168.29.134	192.168.29.129			ICMP	98	Echo (ping) re
36	2014-10-28							
37	2014-10-28							
38	2014-10-28							
39	2014-10-28							
40	2014-10-28							
41	2014-10-28							
42	2014-10-28							
43	2014-10-28							
44	2014-10-28							
45	2014-10-28							
46	2014-10-28							

Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	242	100.00 %	21603	0.002	0	0	0.000
Ethernet	100.00 %	242	100.00 %	21603	0.002	0	0	0.000
Address Resolution Protocol	3.31 %	8	1.56 %	336	0.000	8	336	0.000
Internet Protocol Version 4	96.69 %	234	98.44 %	21267	0.002	0	0	0.000
Internet Control Message Protocol	74.38 %	180	81.66 %	17640	0.002	180	17640	0.002
Transmission Control Protocol	22.31 %	54	16.79 %	3627	0.000	30	1996	0.000
Data	9.92 %	24	7.55 %	1631	0.000	24	1631	0.000

ICMPとTCP  
TCPデータ部分を見てみよう

File: "E:\User\taka\SkyDrive\ドキュメン... Packets: 242 · Displayed: 242 (100.0%) · Load time: 0:00.002 Profile: Default

# TCPのデータ部分を見る

network100.pcap [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

Filter: tcp.stream eq 0

Stream Content:

ctf4b{netcat-is-useful}

Follow TCP Streamすると、  
ctf4b{netcat-is-useful}

Entire conversation (47 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

File: "E:\User\taka\SkyDrive\ドキュメン..." Packets: 242 · Displayed: 54 (22.3%) · Load time: 0:00.003 Profile: Default

# 本日の内容

---

1. ネットワーク分野で必要な知識は？
  - パケット・通信プロトコル
  - Wiresharkの基本的な使い方
2. CTFにおけるネットワーク問題
  - どのような問題が出るのか
3. 問題を見てみよう
  - 問題解くときに見るべきポイント
4. 今後のレベルアップするためには

# 今後レベルアップしていくためには

---

- 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
- 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
- 過去問題の解法を知る = Writeupを読む

# 今後レベルアップしていくためには

---

- 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



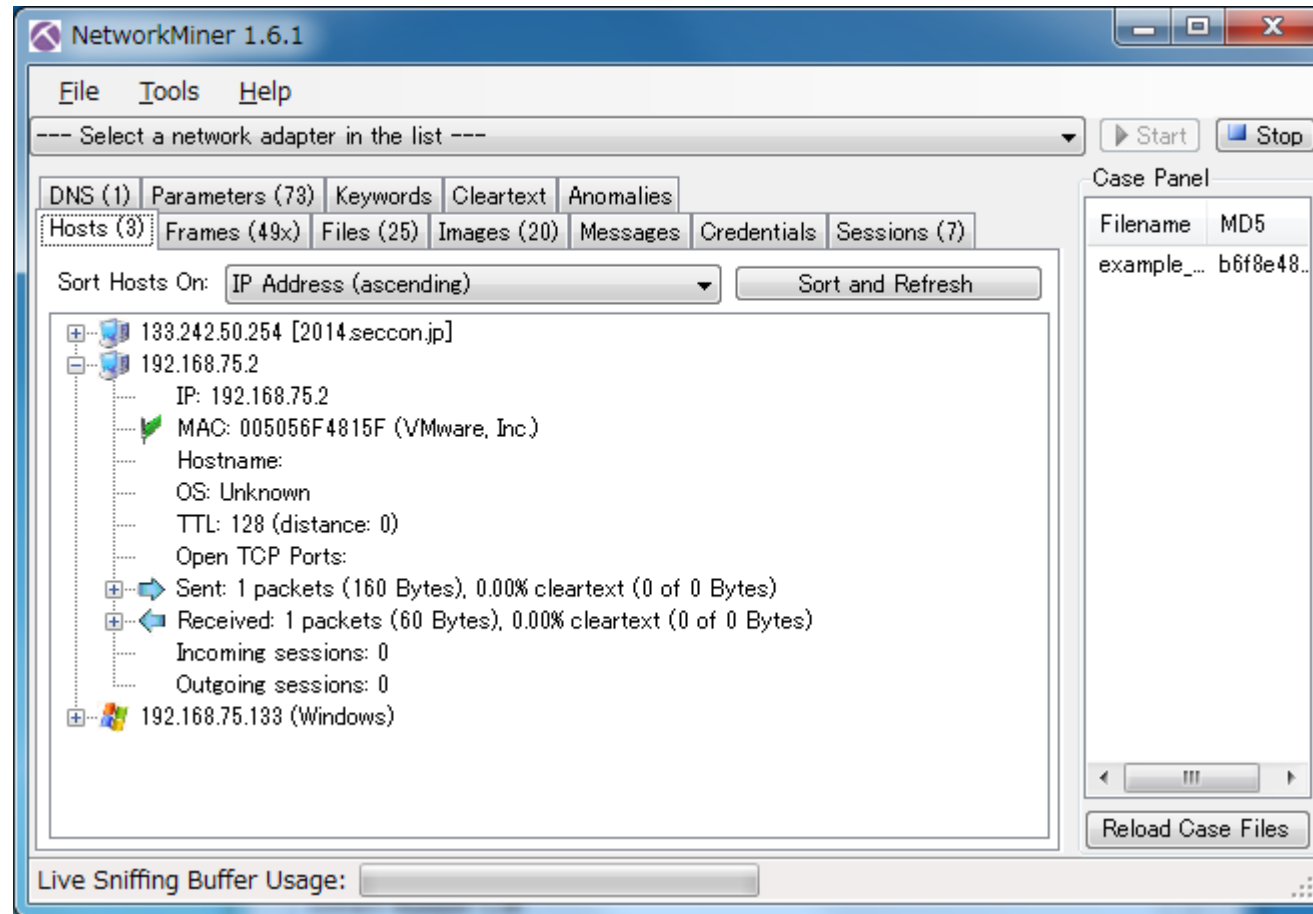
# 付録 1

---

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 information. The 'Open TCP Ports' section is highlighted, listing 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 (Sntp) 23 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	29
2730	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	HEAD.html	html	29
2708	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	index.html	html	89
2900	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	favicon.ico.html	html	29
3052	C:\User...	192.168...	TCP 80	192.168...	TCP 12...	HttpGet...	index[1].html	html	89
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	aus3.mozilla.or...	cer	1 29
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	Thawte SSL C...	cer	1 19
3654	C:\User...	63.245.2...	TCP 443	192.168...	TCP 13...	TlsCerti...	thawte Primar...	cer	1 09
3854	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	index.html	html	24 69
3881	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	styles.css	css	10 19
3910	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	ic_shuryo_s.gif	gif	29
3912	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	ic_new_s.gif	gif	29
3938	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	kddi.png	png	6 89
3952	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	dwango.png	png	4 99
3970	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	isc2.png	png	7 99
3986	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	jinsoken.png	png	6 09
4005	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	temptech.png	png	7 29
4015	C:\User...	133.242...	TCP 80	192.168...	TCP 13...	HttpGet...	jipdec.png	png	3 09

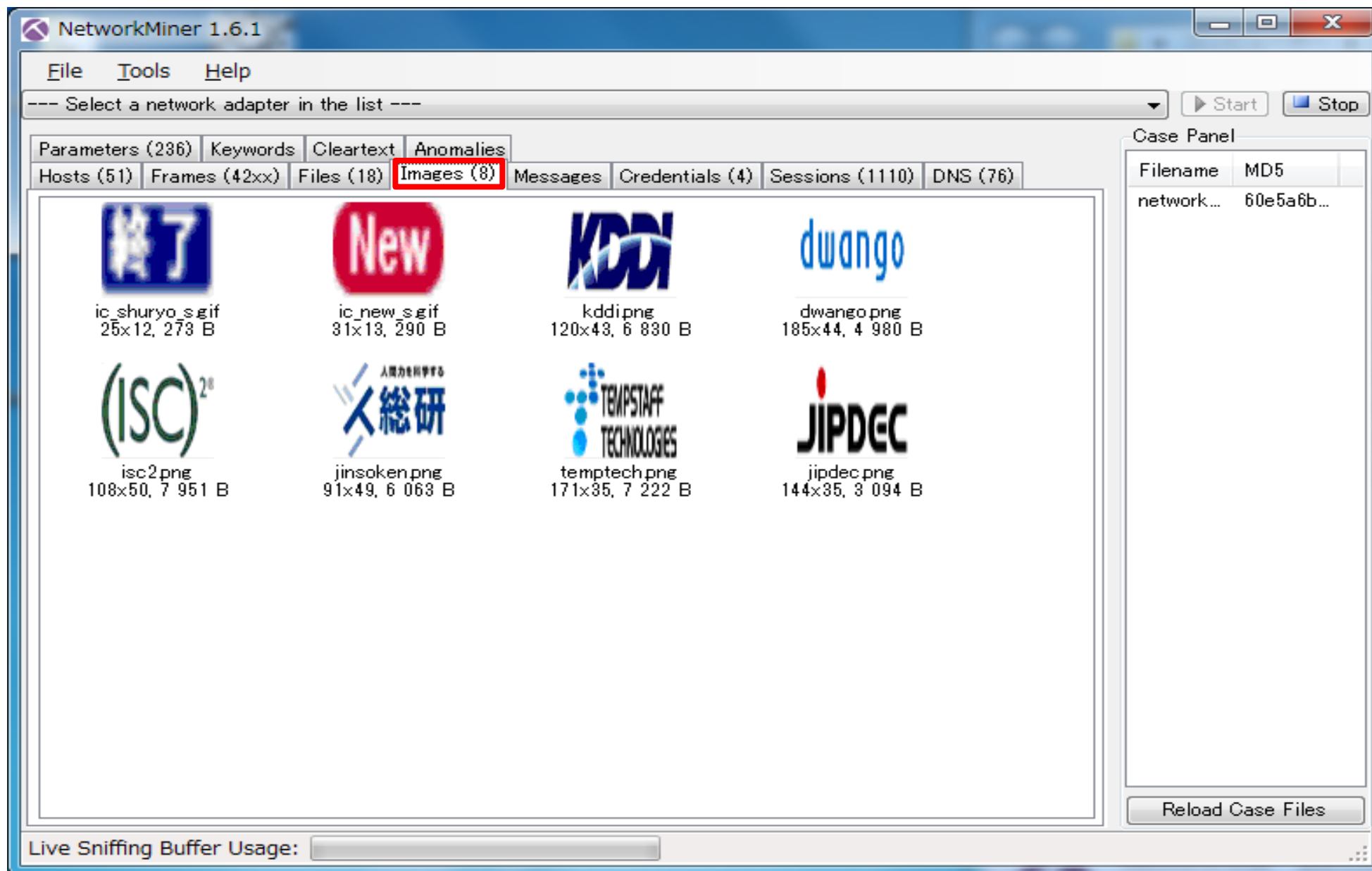
Case Panel

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 and highlighted with a red rectangle. Below the tabs, there are checkboxes for 'Show Cookies', 'Show NTLM challenge-response', and 'Mask Passwords'. The main area displays a table of captured credentials. The 'Case Panel' on the right shows the filename 'network...' and its MD5 hash '60e5a6b...'. At the bottom, there is a 'Live Sniffing Buffer Usage' progress bar.

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...



# 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
  - Wiresharkを使うより効率が良い場合がある
- 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

---

CTFで使えるネットワーク系ツール

# CTFで使えるネットワーク系ツール

---

- CTFで使えると便利なネットワーク系ツール
- 下記のジャンルに分類
  - パケットキャプチャ
  - パケット解析
  - パケット生成・送信
  - Wireshark付属
  - その他
- 赤字のツールは重要 (だと考えてる)ツール

# パケットキャプチャツール

---

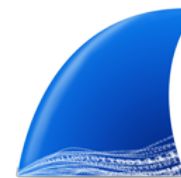
- tcpdump (Linux, Mac OS XなどのUNIX系OS)
  - CUIでパケットキャプチャ、pcapファイルを簡易表示することも
  - 軽い
  - <http://www.tcpdump.org/>
- Windump (Windows)
  - tcpdumpのWindows移植版
  - <http://www.winpcap.org/windump/>
- dumpcap / tshark / Wireshark
  - Wiresharkをダウンロードすると付属
  - dumpcapとtsharkはCUI, WiresharkはGUI
  - 機能的には、dumpcap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