

# Wiresharkの使い方

電子情報通信プログラム  
3年 前田英行



# CTFの問題ジャンル

- Reversing
- Pwn
- Crypto
- Web
- Network
- Forensics

大体こんな感じ

# CTFのWeb/Network分野について

## Web

> Webアプリケーションの脆弱性を攻撃しflagを取得する問題

## Network

> ネットワークパケットを解析し、問題によってはサーバに接続してflagを取得する問題

# CTFのWeb/Network分野について

なので、

攻略対象とするサーバやウェブページの構成を把握し、  
攻略のきっかけとなる脆弱性を発見すること大事

→従って、ネットワーク通信への理解はもちろん  
のこと,php やjavascript といった言語への理解も  
必要となる.

CTFのWeb/Network分野について

OSI参照モデルやTCP/IPとかやってきた

今日は、通信のやり取りを見てみよう！

# 今回の目標

- 通信の内容を目でみる



メイン

- Wiresharkとは何かを知る
- Wiresharkで出来ること & そのやり方を知る
- 日常の通信を覗いてみる
- CTFでの使われ方を知る（ハンズオン）

# Wiresharkとは

Wireshark =

ネットワークパケットをキャプチャ  
して分析するツール



無料で、とても簡単に使えるのでメッチャ便利。  
Kali-linuxにも標準で入っているぐらいいろん  
な人が認めるいいツール

# ネットワークパケット

- パケット

⇒ネットワーク経由でやり取りされるデータの塊のこと

1 パケット = 128 バイト

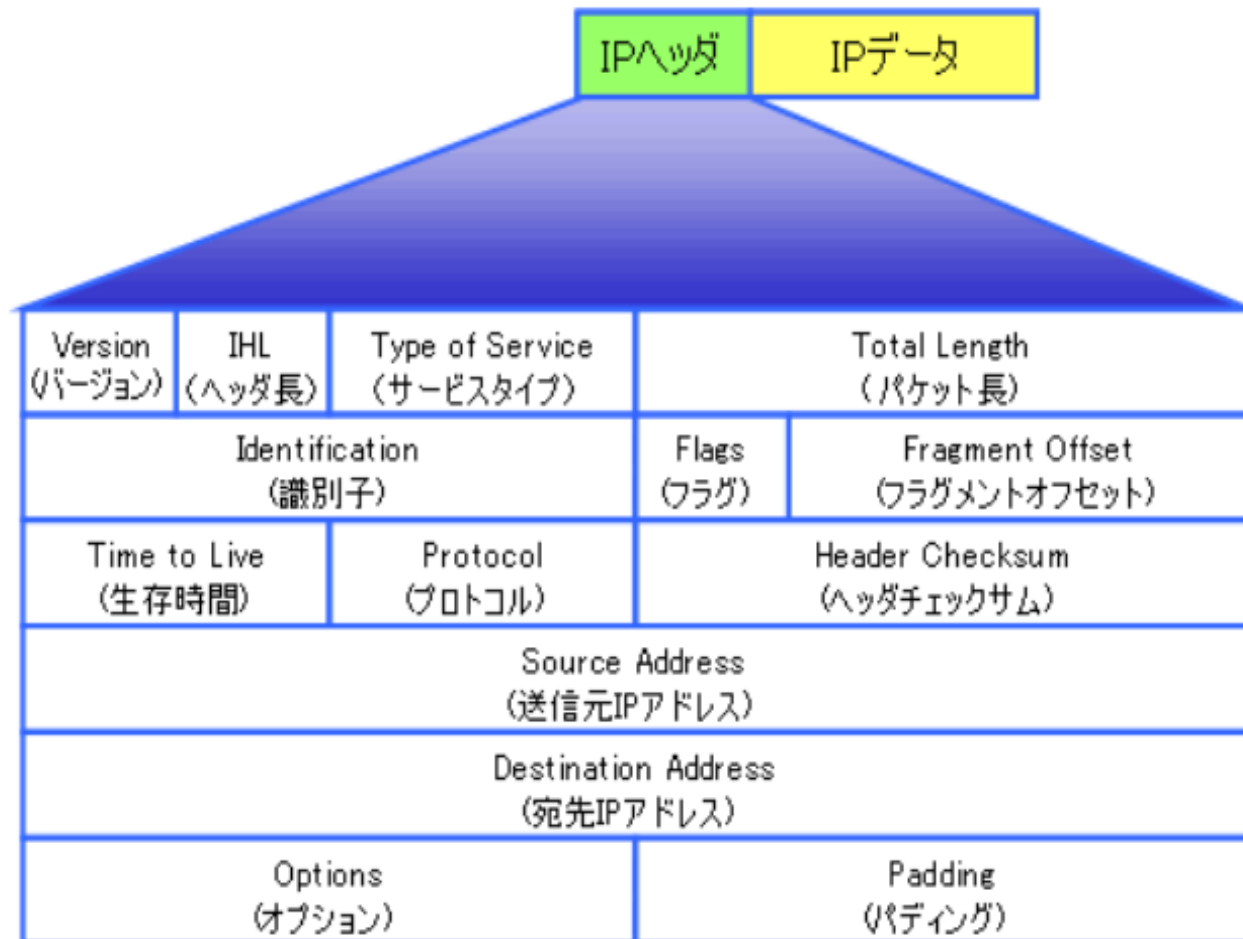
プロトコルに従って分割されたパケット → IP パケット

携帯を契約する時とかに聞いたことがあるかも・・・

通信では、このパケットにデータを分割して行われる



# 1 パケットの中身



IPヘッダ：色々入ってる

IPデータ

：分割された送りたいデータ

他のプロトコルの時は、ヘッダを色々つけることで他のプロトコルヘッダをつける

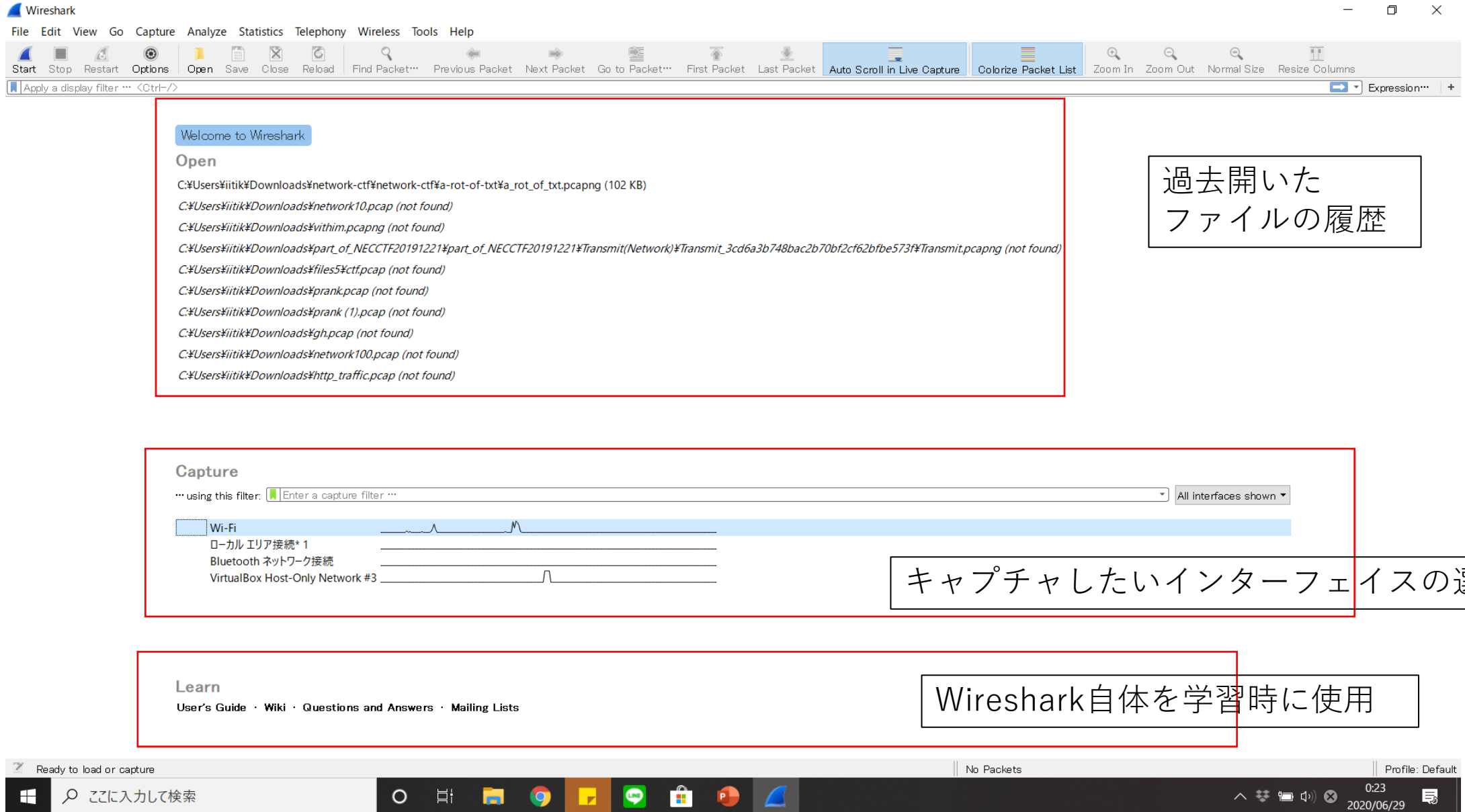
つまり

- Wiresharkは、  
数字の羅列であるパケットの  
やり取りの中身をわかりやすく  
表示してくれる！！

# Wiresharkで出来ること

- パケットのキャプチャ
- パケットをを様々な条件で絞り込み
- パケットの並び替え
- パケットの統計作成
- 送信したファイルの復元
- USBやBluetoothのネットワーク以外のプロトコルも解析できる

# Wiresharkの使い方



The image shows the Wireshark application window with several sections highlighted by red boxes and annotated with Japanese text in white boxes.

- Open Section:** A red box highlights the "Open" section, which lists recent files. A white box to the right contains the text "過去開いたファイルの履歴" (History of files opened).
- Capture Section:** A red box highlights the "Capture" section, showing a list of network interfaces. A white box to the right contains the text "キャプチャしたいインターフェイスの選択" (Selection of the interface to capture).
- Learn Section:** A red box highlights the "Learn" section, which provides links to the User's Guide, Wiki, Questions and Answers, and Mailing Lists. A white box to the right contains the text "Wireshark自体を学習時に使用" (Use Wireshark itself during learning).

The bottom of the window shows the Windows taskbar with the Start button, search bar, and various application icons. The system tray on the right shows the time as 0:23 on 2020/06/29.

# Wiresharkの使い方

①

②

③

④

⑤

Wireshark

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Start Stop Restart Options Open Save Close Reload Find Packet... Previous Packet Next Packet Go to Packet... First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In Zoom Out Normal Size Resize Columns

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.10.6	10.10.10.5	TCP	74	46018 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2686578085 TSecr=0 WS=128
2	0.000057	10.10.10.5	10.10.10.6	TCP	74	21 → 46018 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=454117 TSecr=2686578085
3	0.000364	10.10.10.6	10.10.10.5	TCP	66	46018 → 21 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2686578085 TSecr=454117
4	0.002622	10.10.10.5	10.10.10.6	FTP	108	Response: 220-FileZilla Server version 0.9.46 beta
5	0.002832	10.10.10.5	10.10.10.6	FTP	126	Response: 220-written by Tim Kosse (tim.kosse@filezilla-project.org)
6	0.002864	10.10.10.6	10.10.10.5	TCP	66	46018 → 21 [ACK] Seq=1 Ack=43 Win=64256 Len=0 TSval=2686578088 TSecr=454118
7	0.003073	10.10.10.6	10.10.10.5	TCP	66	46018 → 21 [ACK] Seq=1 Ack=103 Win=64256 Len=0 TSval=2686578088 TSecr=454118
8	0.003112	10.10.10.5	10.10.10.6	FTP	127	Response: 220 Please visit <a href="http://sourceforge.net/projects/filezilla/">http://sourceforge.net/projects/filezilla/</a>
9	0.003393	10.10.10.6	10.10.10.5	TCP	66	46018 → 21 [ACK] Seq=1 Ack=164 Win=64256 Len=0 TSval=2686578088 TSecr=454118
10	0.007351	10.10.10.6	10.10.10.5	FTP	82	Request: USER anonymous
11	0.007873	10.10.10.5	10.10.10.6	FTP	103	Response: 331 Password required for anonymous
12	0.008155	10.10.10.6	10.10.10.5	TCP	66	46018 → 21 [ACK] Seq=17 Ack=201 Win=64256 Len=0 TSval=2686578093 TSecr=454118
13	0.008247	10.10.10.6	10.10.10.5	FTP	83	Request: PASS anonymous@

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: PcsCompu\_e7:aa:cd (08:00:27:e7:aa:cd), Dst: PcsCompu\_81:9b:3f (08:00:27:81:9b:3f)

> Internet Protocol Version 4, Src: 10.10.10.6, Dst: 10.10.10.5

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0xb83c (47164)

> Flags: 0x4000, Don't fragment

Time to live: 64

Protocol: TCP (6)

Header checksum: 0x5a61 [validation disabled]

0000 08 00 27 81 9b 3f 08 00 27 e7 aa cd 08 00 45 00 ..'..?..'. ....E.

0010 00 3c b8 3c 40 00 40 06 5a 61 0a 0a 0a 06 0a 0a ..<.<@.@. Za.....

0020 0a 05 b3 c2 00 15 ed cf 87 38 00 00 00 00 a0 02 .....8.....

0030 fa f0 6e 49 00 00 02 04 05 b4 04 02 08 0a a0 21 ..nI.....!

0040 ed a5 00 00 00 00 01 03 03 07 ..... ..

Ready to load or capture

Packets: 886 · Displayed: 886 (100.0%)

Profile: Default

1:21 2020/06/29

# Wiresharkの使い方

- ①表示フィルタツールバー  
パケットの絞り込みに使用
- ②パケット一覧  
1行が1個のパケットを表す
- ③パケット詳細部  
選択したパケットの詳細が表示
- ④パケット倍列部  
パケットの中身が表示される
- ⑤ステータスバー  
パケットの状態が表示される。コメントも付けられる

# Wiresharkの使い方

## 画面のレイアウト変更

メニューの編集(edit) > 設定(Preference) > 外観 (Appearance)  
> レイアウト

好きなものに変更してください

## Wiresharkの使い方 ②パケット一覧

列の名前	表示内容
No(番号)	パケットの通し番号。要求と応答矢印で表示したりする。
Time(時間)	キャプチャした時刻を表示
Source(送信元)	パケット送信元のアドレス
Destination(宛先)	パケットの宛先アドレス
Protocol(プロトコル)	上位レイヤのアドレス
Length(長さ)	パケットのサイズ
Info	パケットの概要表示



## Wiresharkの使い方 ③パケット詳細部

②のパケット一覧部で選択したパケットの内容をより詳細に表示する。ヘッダやフィールドの値が表示される。

## Wiresharkの使い方 ④パケット倍列部

パケットのアドレス

16進数ダンプ（実際のデータ内容）

ASCII表示（データ内容を文字で表す）

# Wiresharkの使い方   メニューバー

File   Edit   View   Go   Capture   Analyze   Statistics   Telephony   Wireless   Tools   Help

## File

Capture(キャプチャ)    パケットキャプチャの開始・終了・編集を行う。

Analyze(分析)    フィルタの設定やパケット解析を行う

Statistics (統計)    プロトコルごとの統計、分布などの統計情報を表示する

Telephony (電話)    VoIPに関する統計表示

Wireless (無線)

Tools

Wiresharkの使い方 メニューバー > File

File>Object Exprt > HTTP

キャプチャファイルから含まれる文書や画像ファイルを抽出できる

Wiresharkの使い方   メニューバー > File  
  > View

File>Object Exprt > HTTP

HTTPで行われたキャプチャファイルから含まれる文書や画像ファイルを抽出できる

# View > Colorize Conversation

プロトコルの対話部分を色付けしてくれる。

Wiresharkの使い方 メニューバー > Statics

Statics>Protocol Hierarchy（プロトコル分析）  
プロトコル分布をツリー形式で表示。

Statics>Conversations（会話）  
パケットキャプチャを2点間の通信に分類し、バイト数上位から表示

Statics>Endpoints（終端）  
パケットキャプチャを端末ごとに分類し、バイト数上位順から表示

Statics>Packet Length（パケット長）  
パケット長を元に分類し、統計的に表示

## Wiresharkの使い方 メニューバー > Statics

Protocol Hierarchy (プロトコル分析)	プロトコル分布をツリー形式で表示。
Conversations (会話)	パケットキャプチャを2点間の通信に分類し、バイト数上位から表示
Endpoints (終端)	パケットキャプチャを端末ごとに分類し、バイト数上位順から表示
Packet Length (パケット長)	パケット長を元に分類し、統計的に表示
I/O Graph(入出力グラフ)	送受信の通信料をグラフで表示。
Service Response Time (サービス応答時間)	リクエストとそのレスポンス間に要した時間で表示

## Wiresharkの使い方 ①表示フィルタツールバー

表示フィルタ

ip.addr== 10.255.255.255

(例 ipアドレスで絞り込みをかけたいとき)



## Wiresharkの使い方

**Wiresharkの使い方はこれで終了**

**ただ、説明していないことも多いので、一通り慣れて暇だったら他の機能も調べてみてください・・・**

日常の通信を覗いてみる

## 最初に注意点

以外に重いので、他のアプリ使ってたら終了しといてください  
通信してそうなアプリは、出来る限り切る。

# 日常の通信を覗いてみる パケットキャプチャを開始する

## 手順

- 1.メニューバーのキャプチャ>オプションを押す
- 2.入力欄でキャプチャしたいインターフェイスを選ぶ  
(通信を行っているものを選ぶ)
- 3.出力欄でパケットの保存する先を選ぶ
- 4.Webブラウザを起動して、好きなホームページを閲覧する。

実際にやってみよう！！

Wiresharkを使ったCTF    Seccon for Begginers    sample.pcap

## 問題

1つだけ認証が通っている通信があるのでその時のpasswordを探してください

# Wiresharkを使ったCTF    Seccon Begginers    sample.pcap

パケットを眺めてみる

30	21.267225	127.0.0.1	127.0.0.1	TCP	56 [TCP Window Update] 80 → 50955 [ACK] Seq=1 Ack=1 Win=408288 Len=0 TSval=519241301 TSecr=519241301
31	21.267330	127.0.0.1	127.0.0.1	HTTP	742 POST /ctf_web/login/index.php HTTP/1.1 (application/x-www-form-urlencoded)
32	21.267362	127.0.0.1	127.0.0.1	TCP	56 80 → 50955 [ACK] Seq=1 Ack=687 Win=407584 Len=0 TSval=519241301 TSecr=519241301
33	21.273358	127.0.0.1	127.0.0.1	HTTP	1120 HTTP/1.1 200 OK (text/html)
34	21.273385	127.0.0.1	127.0.0.1	TCP	56 50955 → 80 [ACK] Seq=687 Ack=1065 Win=407232 Len=0 TSval=519241307 TSecr=519241307
35	26.275067	127.0.0.1	127.0.0.1	TCP	56 80 → 50955 [FIN, ACK] Seq=1065 Ack=687 Win=407584 Len=0 TSval=519246295 TSecr=519241307

POSTメソッド    &    200    OK

/ctf\_web/login/index.php

なんか認証してるみたい

こんな通信ばかり

Wiresharkを使ったCTF    Seccon for Begginers    sample.pcap

フィルタ機能を使ってみる。

表示フィルタのところで「HTTP」と入力

1つずつ見てると

HTTP/1.1 302 Found  
/ctf\_web/login/main.php

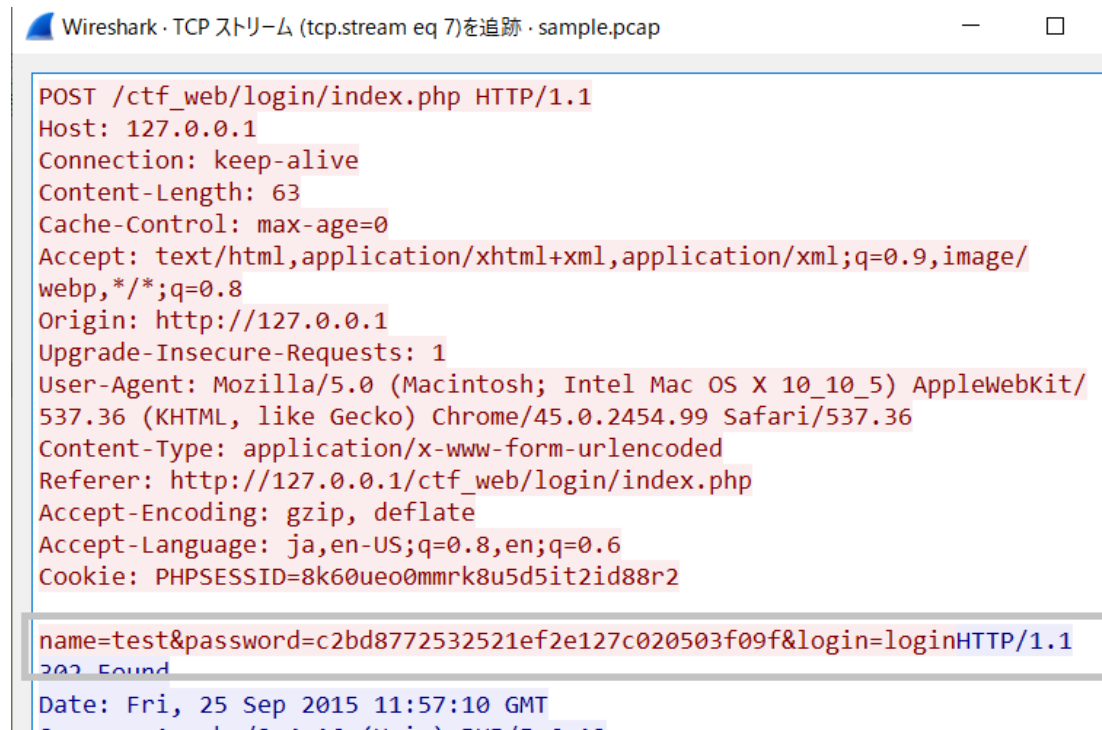
/ctf\_web/login/logout.php

ログインに成功してるみたい

# Wiresharkを使ったCTF Seccon Begginers sample.pcap

302 Found の送受信されたTCPデータを見してみる

302 Foundの packets 上で左クリック  
Follow(追跡) > TCPストリーム



The image shows a screenshot of the Wireshark TCP Stream window. The title bar reads "Wireshark · TCP ストリーム (tcp.stream eq 7)を追跡 · sample.pcap". The main content area displays the raw data of the TCP stream, which is an HTTP 302 Found response. The status line "POST /ctf\_web/login/index.php HTTP/1.1" is highlighted in red. The response body, which is the redirect URL, is highlighted in blue and contains the text "name=test&password=c2bd8772532521ef2e127c020503f09f&login=loginHTTP/1.1". Below the raw data, the packet list shows "302 Found" and "Date: Fri, 25 Sep 2015 11:57:10 GMT".

```
POST /ctf_web/login/index.php HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Content-Length: 63
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://127.0.0.1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.99 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://127.0.0.1/ctf_web/login/index.php
Accept-Encoding: gzip, deflate
Accept-Language: ja,en-US;q=0.8,en;q=0.6
Cookie: PHPSESSID=8k60ueo0mmrk8u5d5it2id88r2

name=test&password=c2bd8772532521ef2e127c020503f09f&login=loginHTTP/1.1
302 Found
Date: Fri, 25 Sep 2015 11:57:10 GMT
```

Wiresharkを使ったCTF    Seccon Begginers    sample.pcap

今回のFlag

c2bd8772532521ef2e127c020503f09f



# Wiresharkを使ったCTF    Seccon Begginers    sample.pcap

問題を解く際の注目するところ

怪しい通信はないか？

例、認証がそこだけ通ってるor通ってない

データ量が多い

プロトコルが違う

などなど

# Wiresharkを使ったCTF    Seccon Begginers    sample.pcap

今回の問題は、通信の内容が簡単にわかった

それは、暗号化するプロトコルを使ってないから  
暗号化しないプロトコル=http,FTP,telnet,smtp

暗号解読はcrypto分野なのでそっちに任せましょう

# Wiresharkを使ったCTF    Seccon Begginers    sample3.pcap

データのやり取りするのはプロトコルを意識する  
(FLAGは基本データなので)  
例、TCP、FTP、SMTP

暗号化されていないプロトコルを見る

プライベートIP同士のやり取り  
→作問者が意図的に発生させた可能性あり

ポート番号を意識する  
→プロトコル毎に使うポート番号は一般的に決まってるので、それ以外が使われてたら怪しい

## Wiresharkの使い方 ④パケット倍列部

パケットのアドレス

16進数ダンプ（実際のデータ内容）

ASCII表示（データ内容を文字で表す）

# 参照サイト

## 1 パケットの中身

<https://www.itbook.info/study/p87.html>

WiresharkのWikipedia

<https://wiki.wireshark.org/>

いろんなパケットキャプチャのデータが記録されてるなど、便利な情報がいろいろある

問題の解説

<https://www.slideshare.net/ctf4b/ctf-for-60147258?ref=https://kyonta1022.hatenablog.com/entry/2018/05/14/003422>

では、次は実際に問題をみんなで  
やってみましょう