SharkFest '16 Europe

Wireshark as a Spy Watermark Pen: Decrypting and Retrieving Information from Packets

16 Oct. 2016

#sf16eu

Megumi Takeshita
Packet Otaku, ikeriri network service co.,ltd.

Megumi Takeshita, ikeriri network service

- Reseller of Riverbed Technology (former CACE technologies) and Metageek, Dualcomm, Profitap and capture products in Japan
- Wrote 10+ books of Wireshark and capturing and network analysis.
- Attending all Sharkfest and translator of QT Wireshark into Japanese

Wireshark as a Spy Watermark Pen

Decrypting and Retrieving Information from Packets

- Wireshark is a good tool that provides relevant information from packets.
- In this session, demonstrate useful 8 cases of decrypting and retrieving information from packets using Wireshark, including wireless (WEP/WPA2), SSL/TLS, HTTP/SMB/TFTP, raw data, and more.

Wireshark is almighty decorder

- Decrypting and retrieving information from packet
  1. Decrypting WEP/WPA2 data
  2. Decrypting TLS/SSL data with key pair
  3. Dectypting TLS/SSL data without key pair.
  4. Retrieving Unicode Characters
  5. Retrieving object files (HTTP/TFTP/SMB)
  6. Retrieving values of field
  7. Retrieving JSON data
  8. Wireshark is the source of big data analysis !

1. Decrypting WEP/WPA2 data

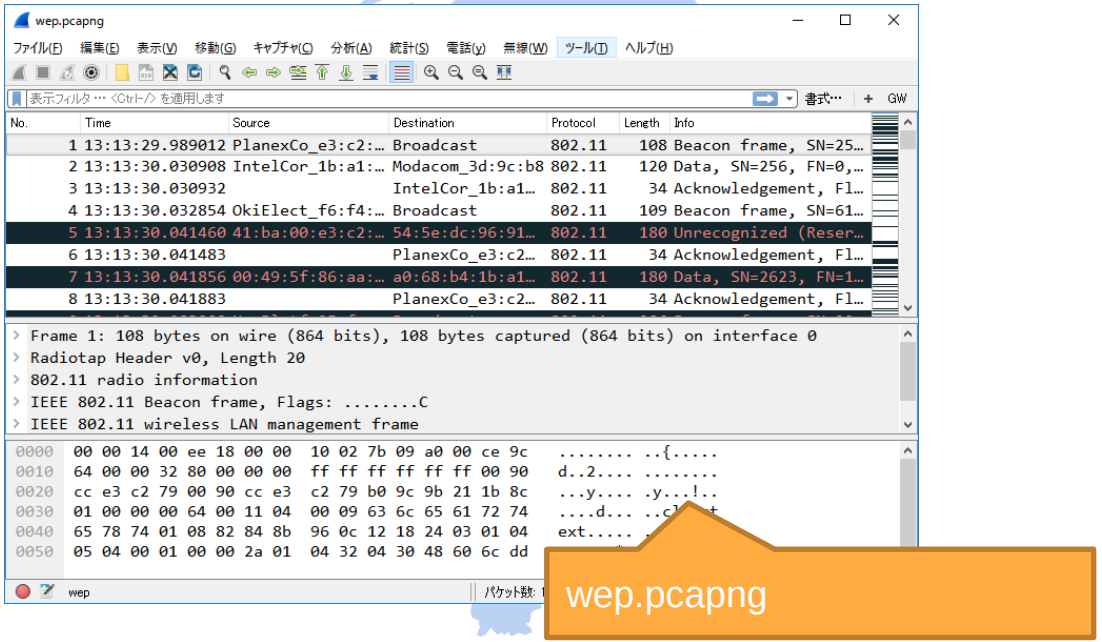
1. Decrypting WEP/WPA2 data

- WEP decryption works well only with the key
  1. Capture packets using AirPcap or monitor mode driver
  2. Set WEP key in IEEE802.11 preference
- WPA2 decryption needs full 4 way handshake
  1. Capture packets using AirPcap or monitor mode driver
  2. Check complete 4 way handshake with eapol filter
  3. Set SSID and passphrase in IEEE802.11 preference

1. Decrypting WEP/WPA2 data

1. Capture packets using AirPcap or monitor mode driver

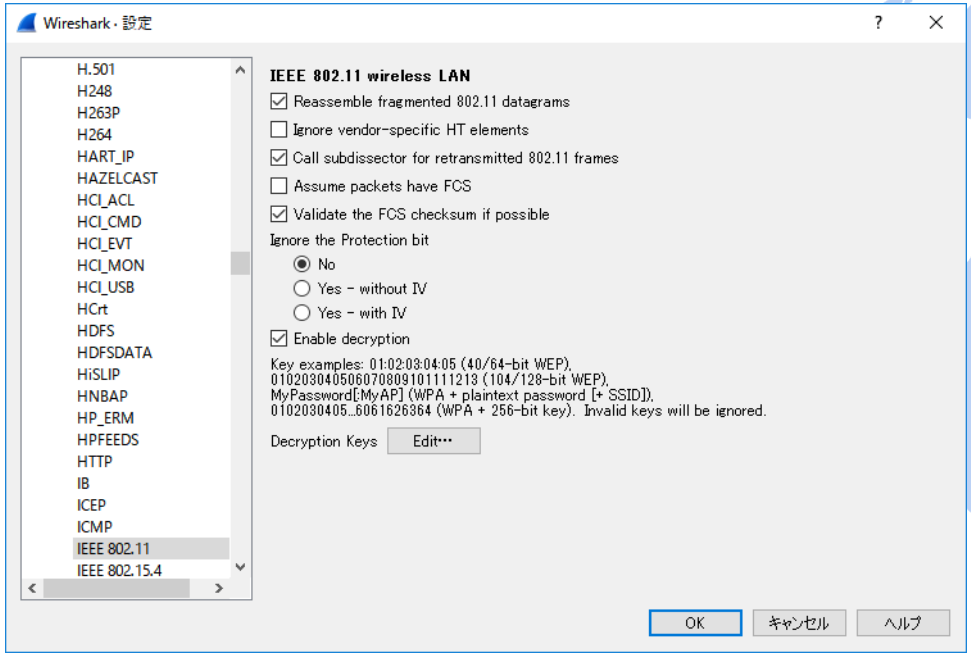
wep.pcapng(http://www.ikeriri.ne.jp/temp/)

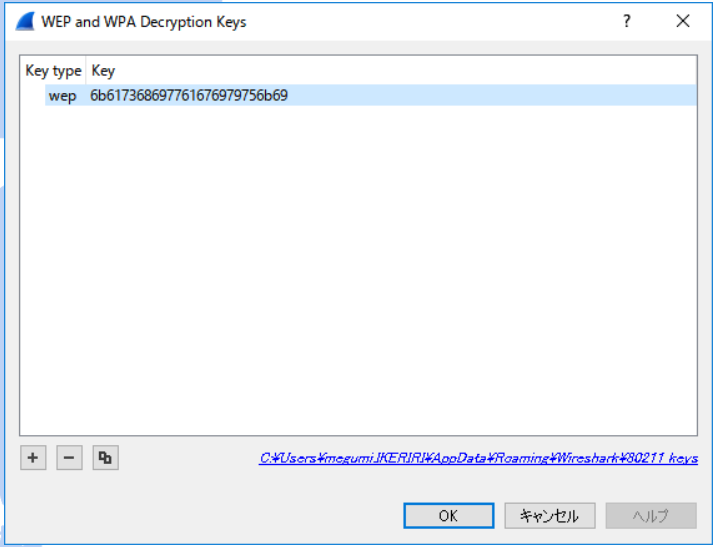


1. Decrypting WEP/WPA2 data

2. Set WEP key in IEEE802.11 preference

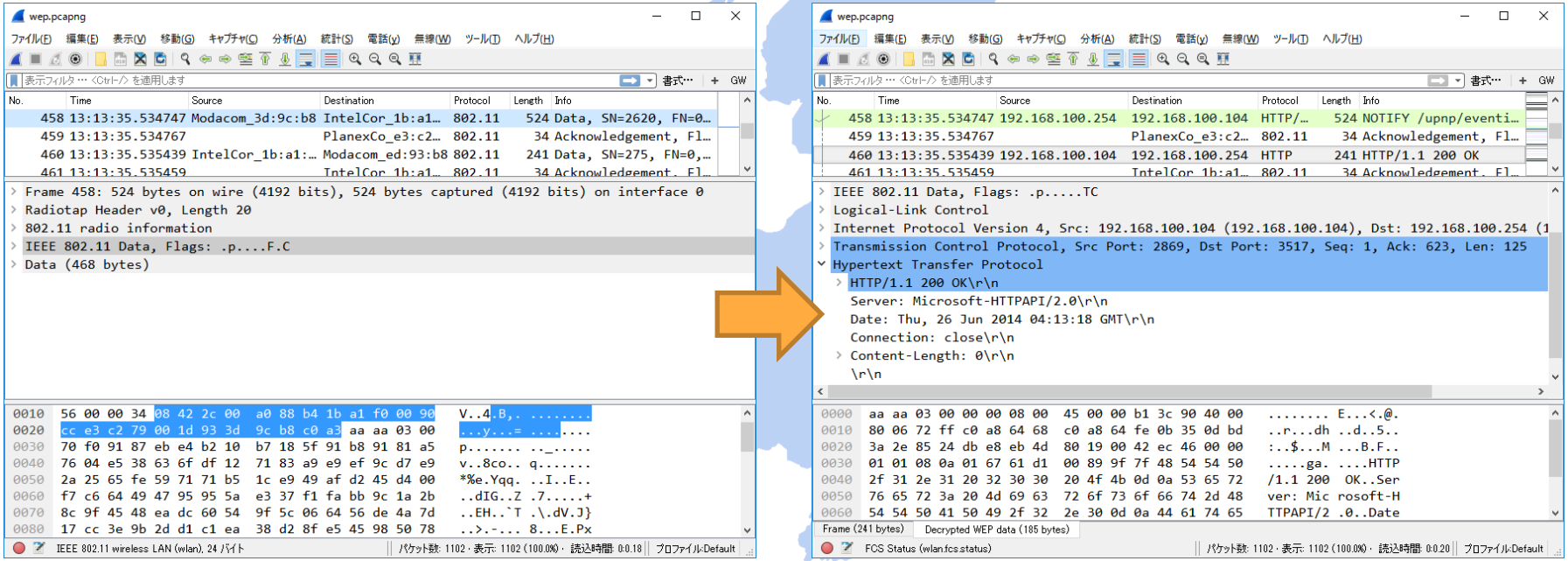
WEP key : kashiwagiyuki (6b6173686977616769756b69)





1. Decrypting WEP/WPA2 data

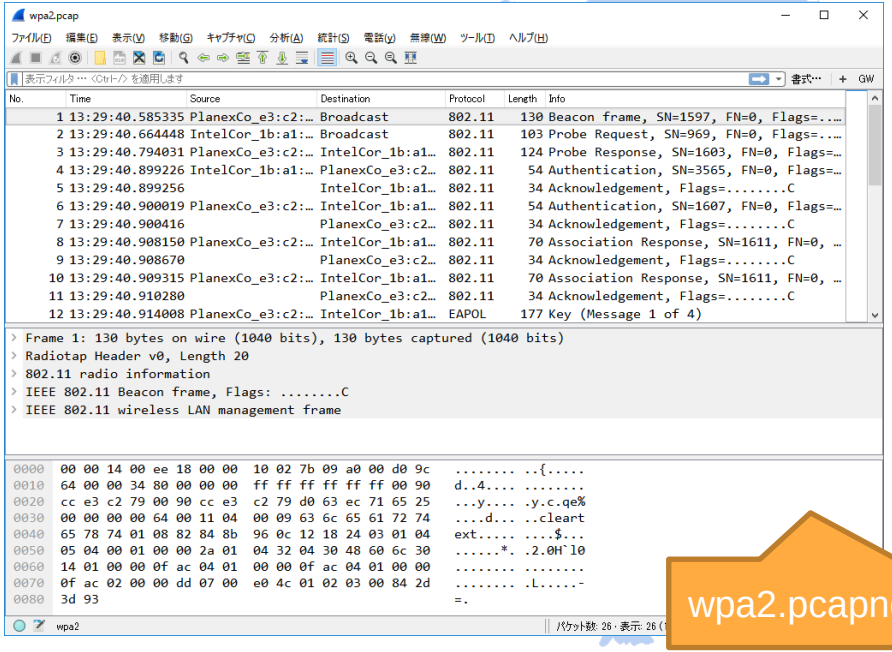
Check #458 in trace file wep.pcapng



1. Decrypting WEP/WPA2 data

1. Capture packets using AirPcap or monitor mode driver

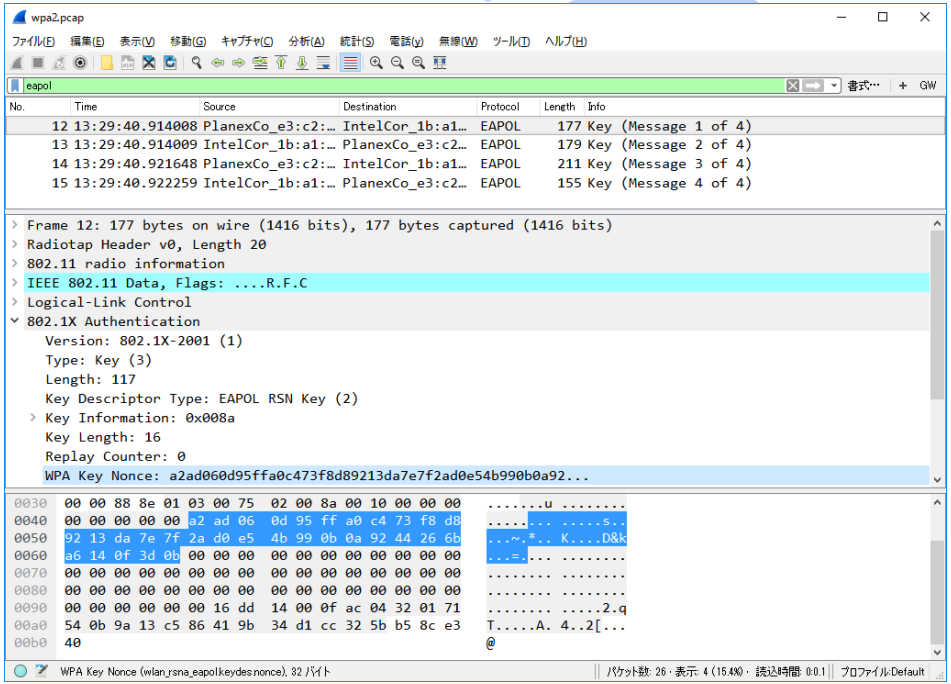
wpa2.pcapng(http://www.ikeriri.ne.jp/temp/)



1. Decrypting WEP/WPA2 data

2. Check complete 4 way handshake with eapol filter

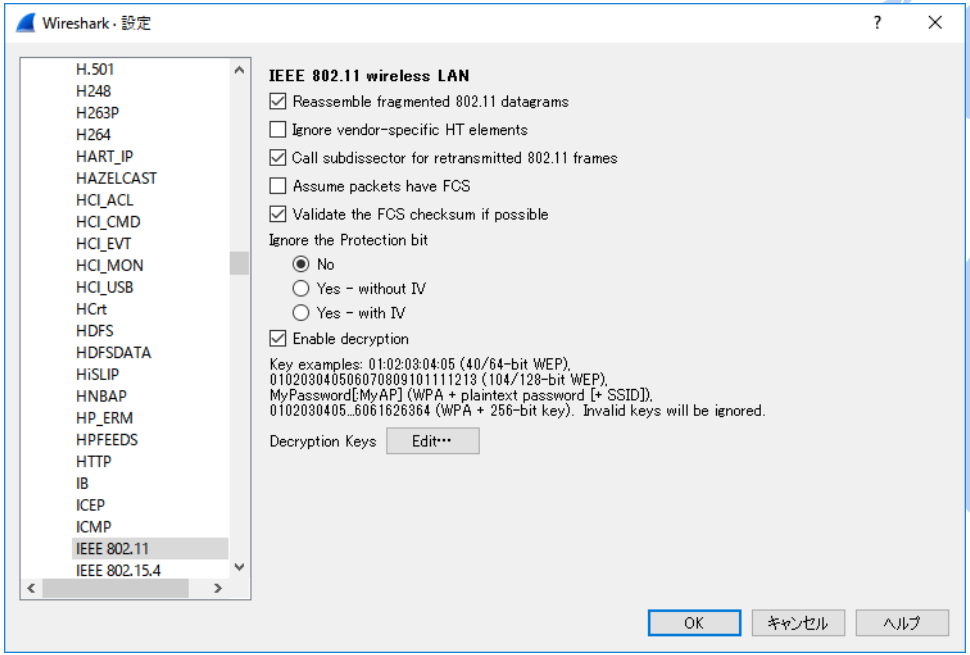
Note: Trace file must contain complete set of 4 way handshake

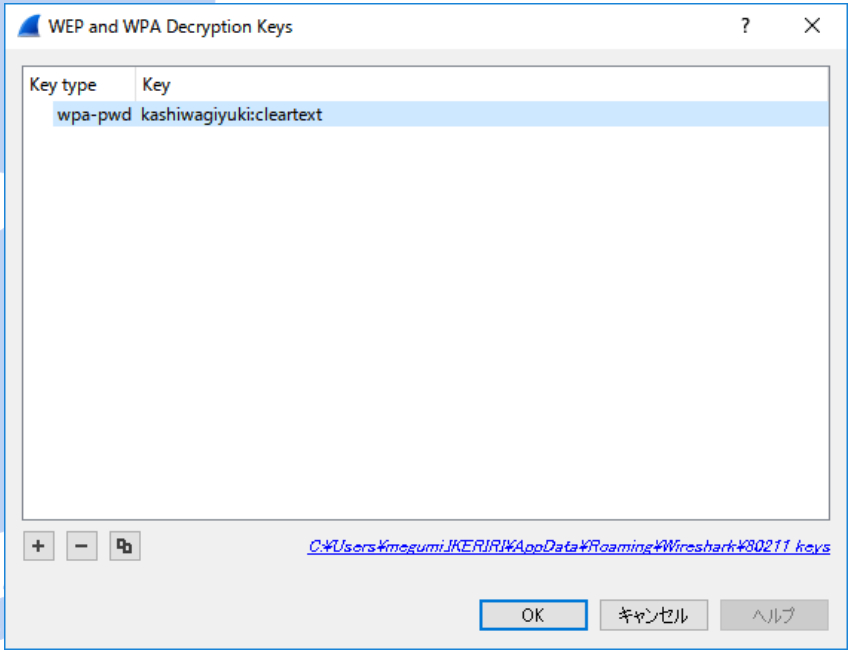


1. Decrypting WEP/WPA2 data

3. Set SSID and passphrase in IEEE802.11 preference

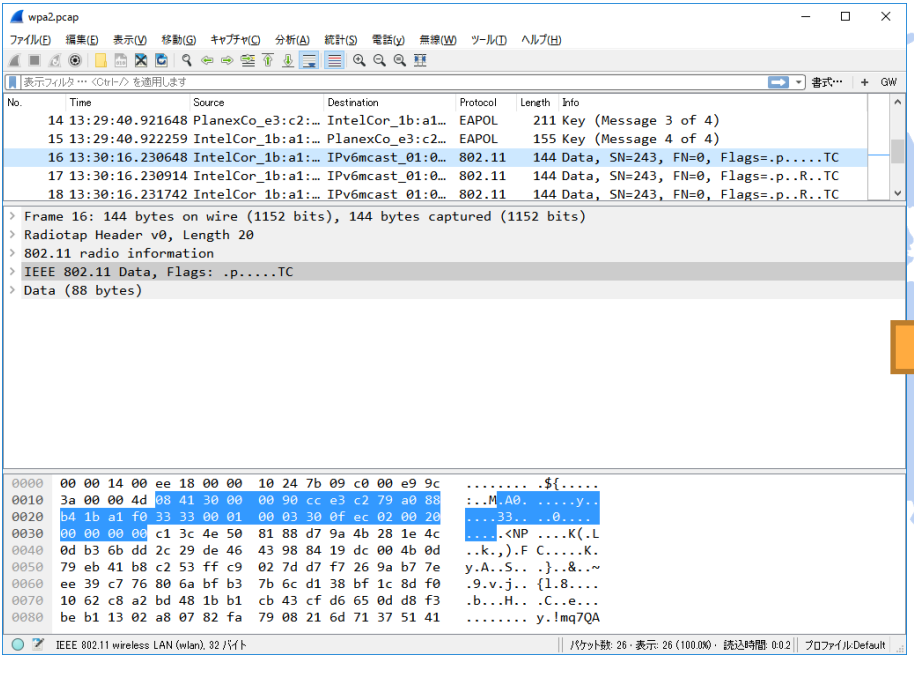
Passphrase : kashiwagiyuki SSID : cleartext

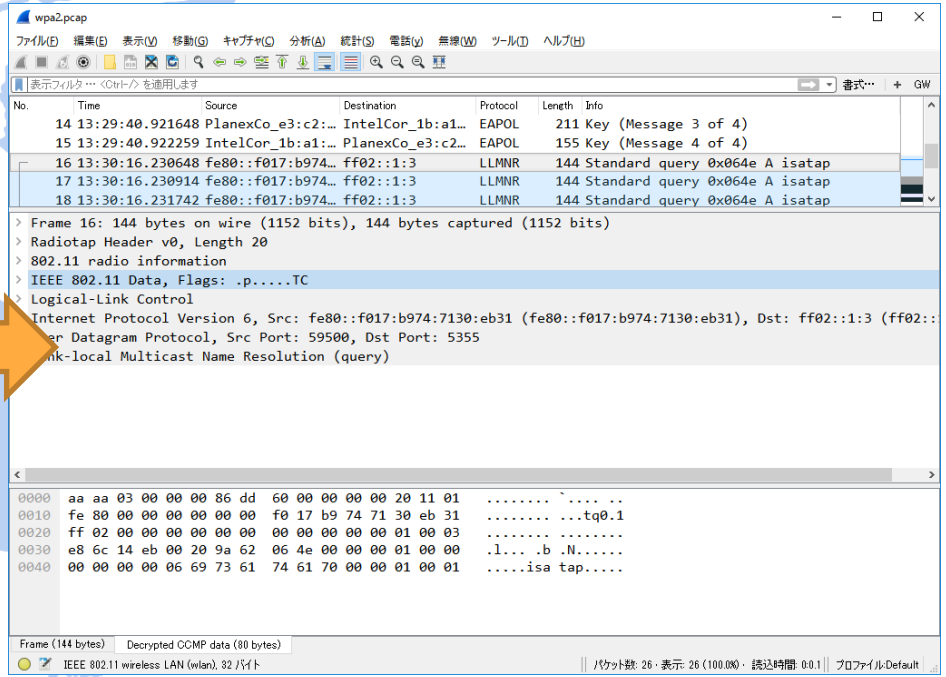




1. Decrypting WEP/WPA2 data

Check #16 in trace file wpa2.pcapng





2. Decrypting TLS/SSL data with key pair

2. Decrypting TLS/SSL data with key pair

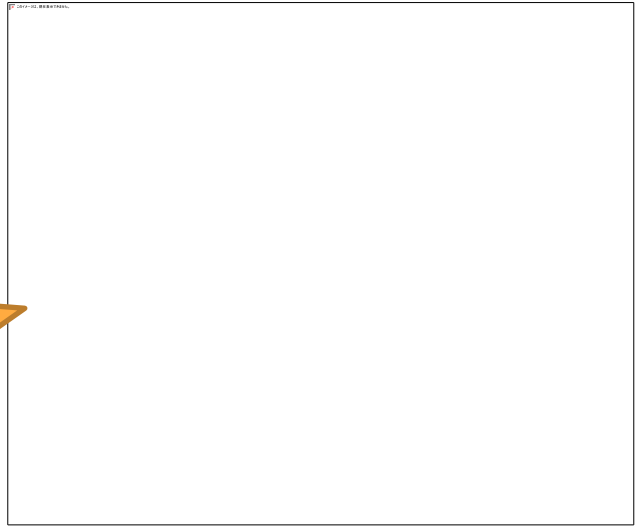
1. PEM format file (certification with public and private key) needed for decryption
   Collect and convert information from server
2. Capture packets including SSL
   Check SSL/TLS handshake in a trace file
3. Set server ip, tcp port, protocol and key(PEM file) in SSL preference

2. Decrypting TLS/SSL data with key pair

1. PEM format file (certification with public and private key) needed for decryption
   Collect and convert information from server

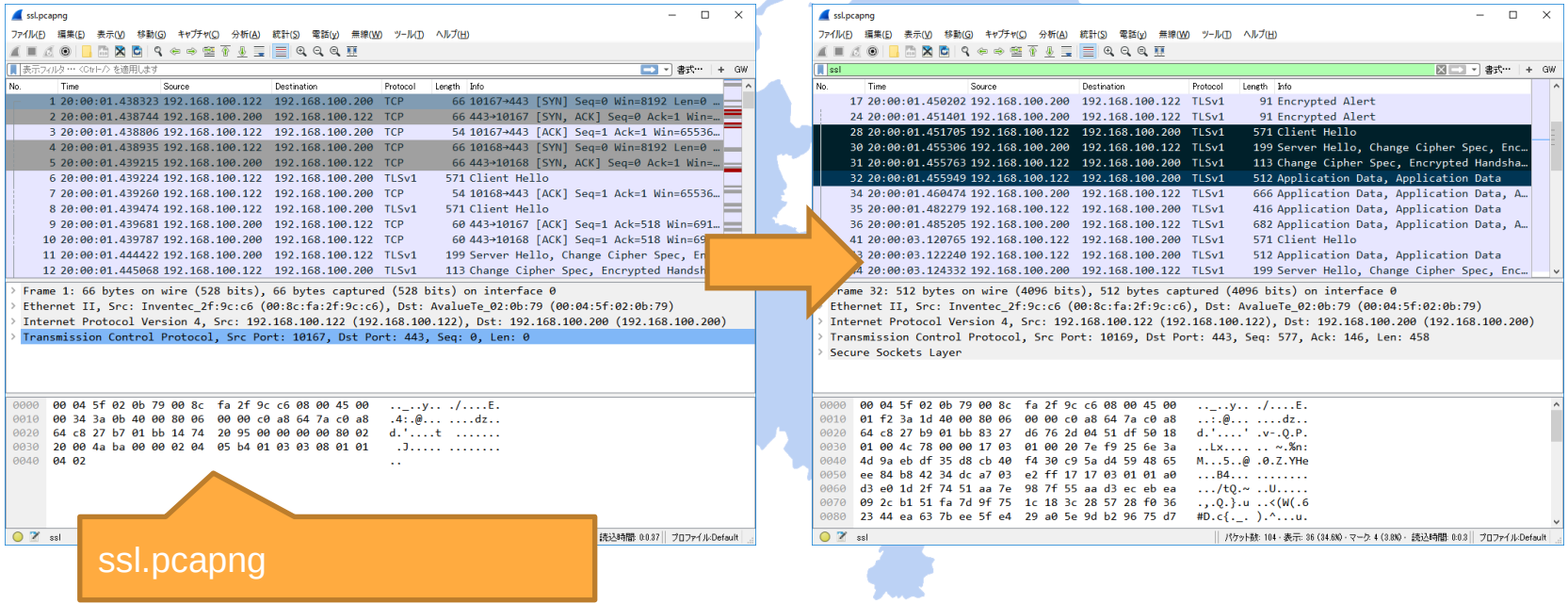
IP address : 192.168.100.200 TCP port : 443
Apache config : /etc/apache2/httpd.conf
SSL config : /etc/apache2/sites-available/httpd-ssl.conf
cert file : /etc/apache2/ssl/cert-file.crt

PEM (Privacy-enhanced Electronic Mail) contains server private and public key



2. Decrypting TLS/SSL data with key pair

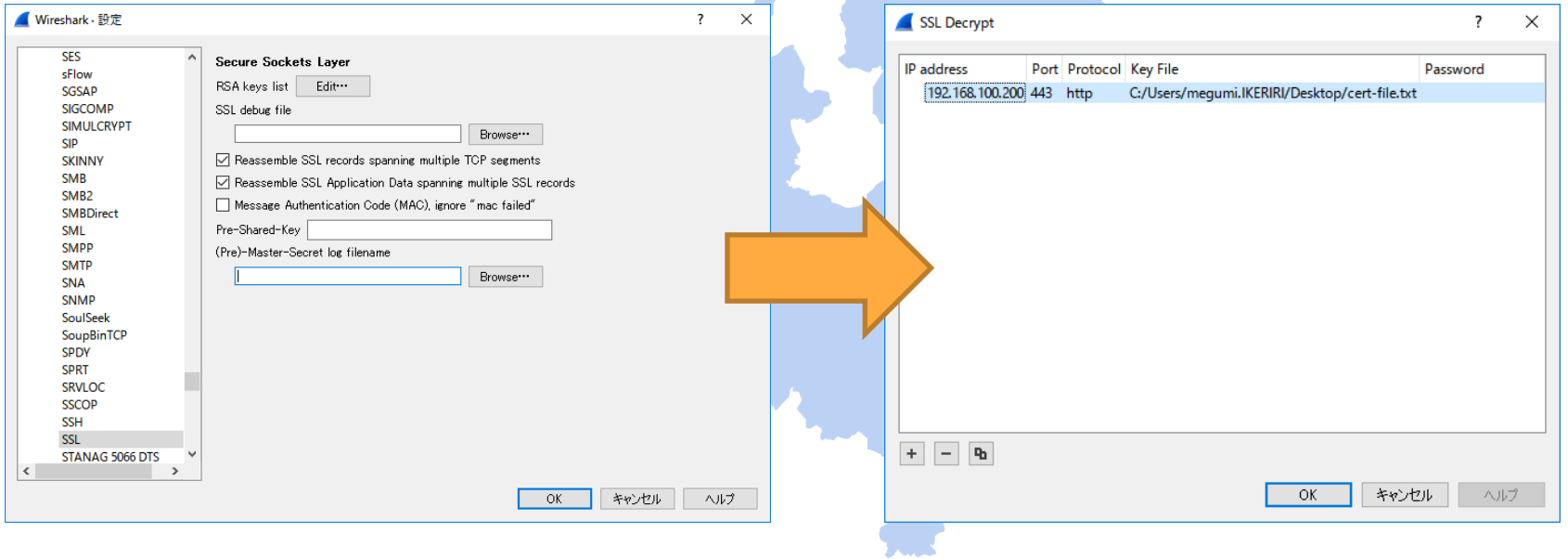
2. Capture packets including SSL (https://192.168.100.200) ssl.pcapng
   Check SSL/TLS handshake in a trace file



2. Decrypting TLS/SSL data with key pair

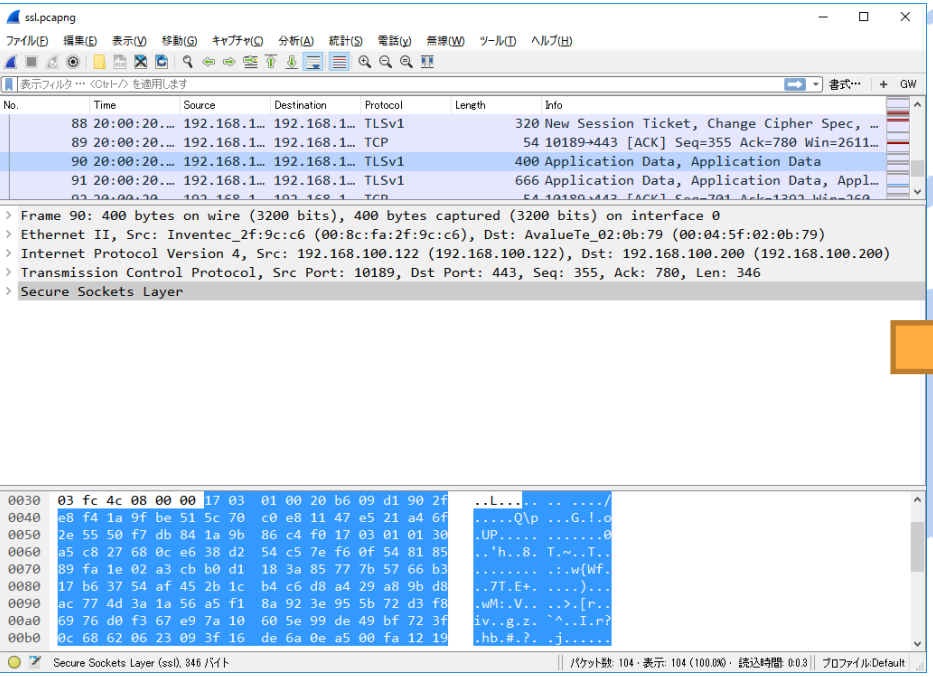
3. Set server ip, tcp port, protocol and key(PEM file) in SSL preference

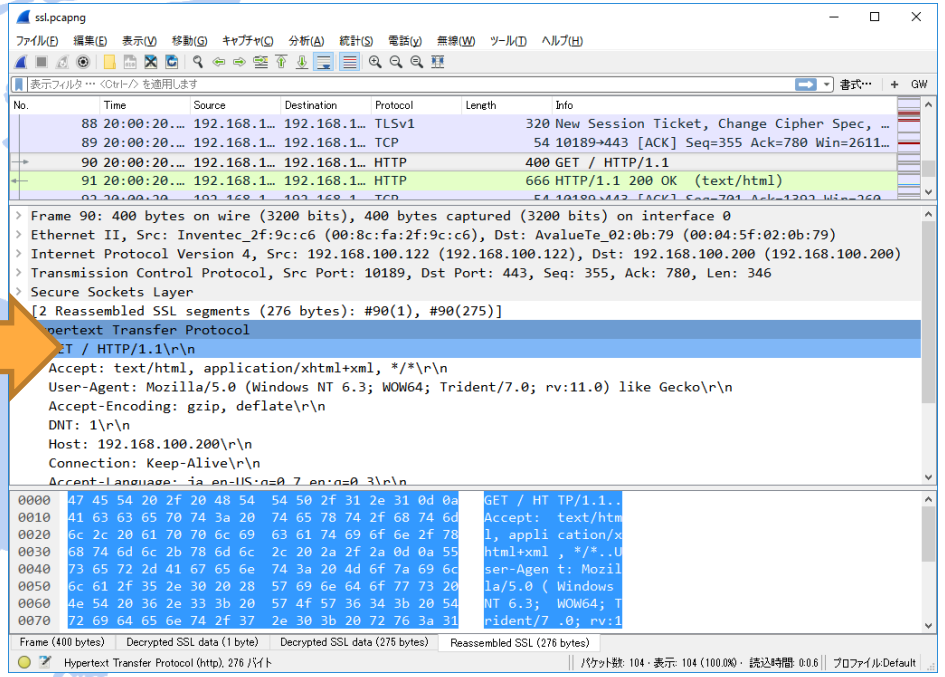
192.168.100.200,443,http,cert-file.txt



2. Decrypting TLS/SSL data with key pair

Check #90,91 in trace file ssl.pcapng

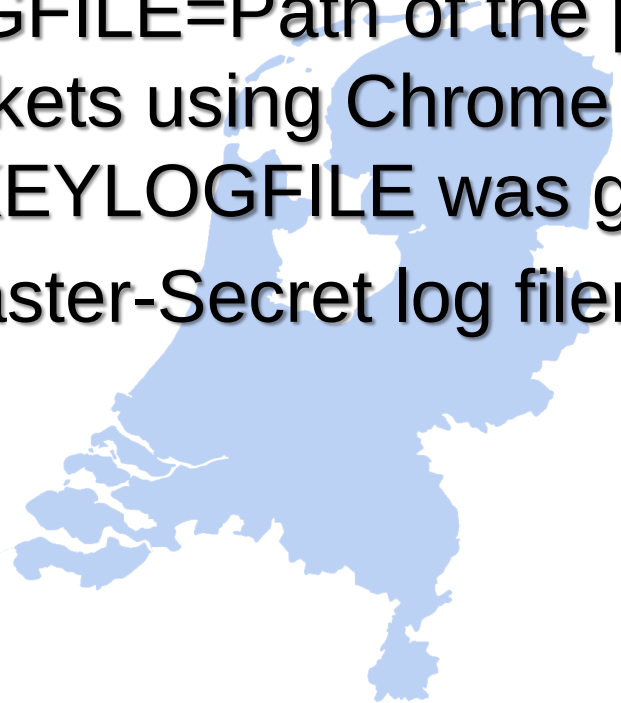




3. Decrypting TLS/SSL data without key pair

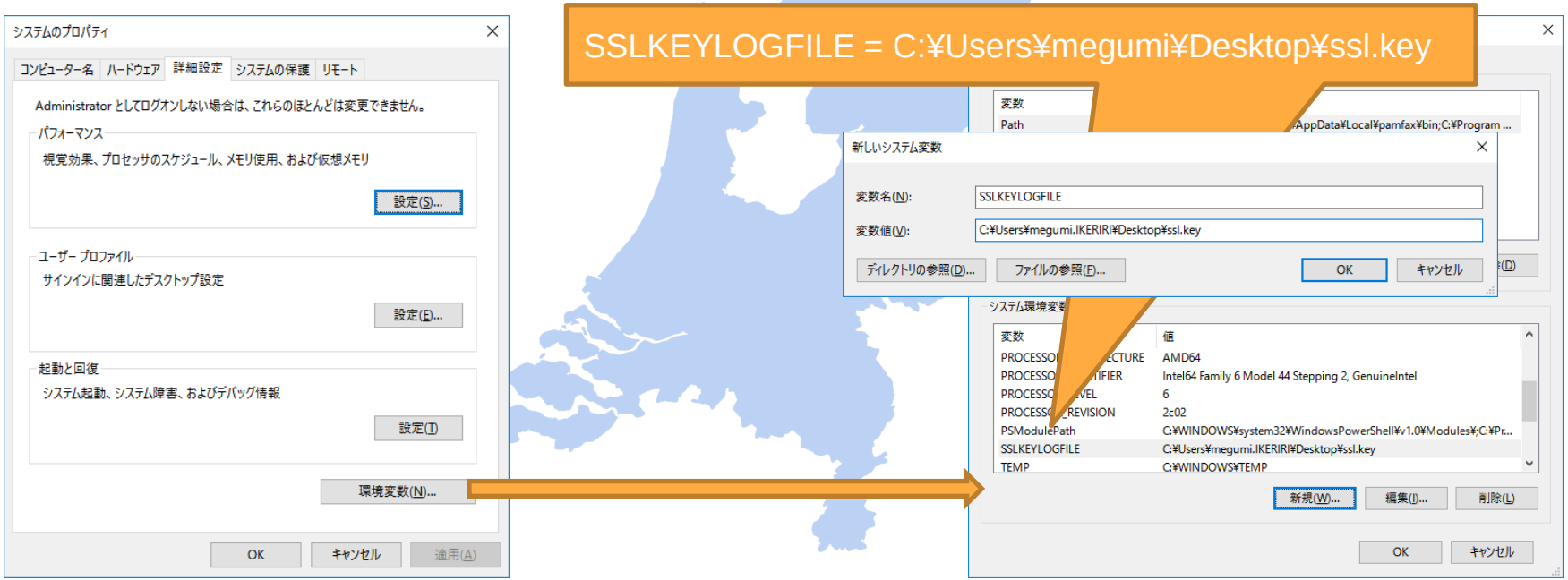
3. Decrypting TLS/SSL data without key pair

1. Set system enviromental variable SSLKEYLOGFILE=Path of the premaster secret
2. Capture packets using Chrome
   Check SSLKEYLOGFILE was generated
3. Set (Pre)-Master-Secret log filename in SSL preference



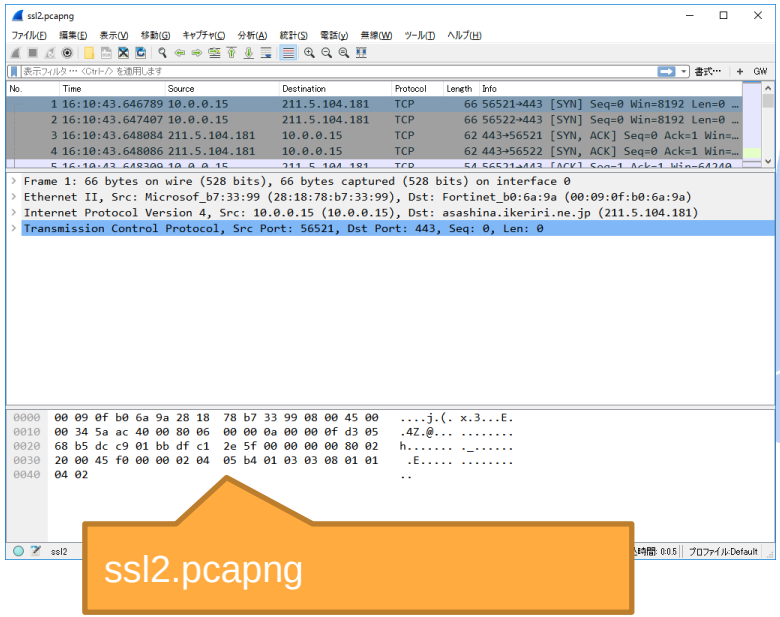
3. Decrypting TLS/SSL data without key pair

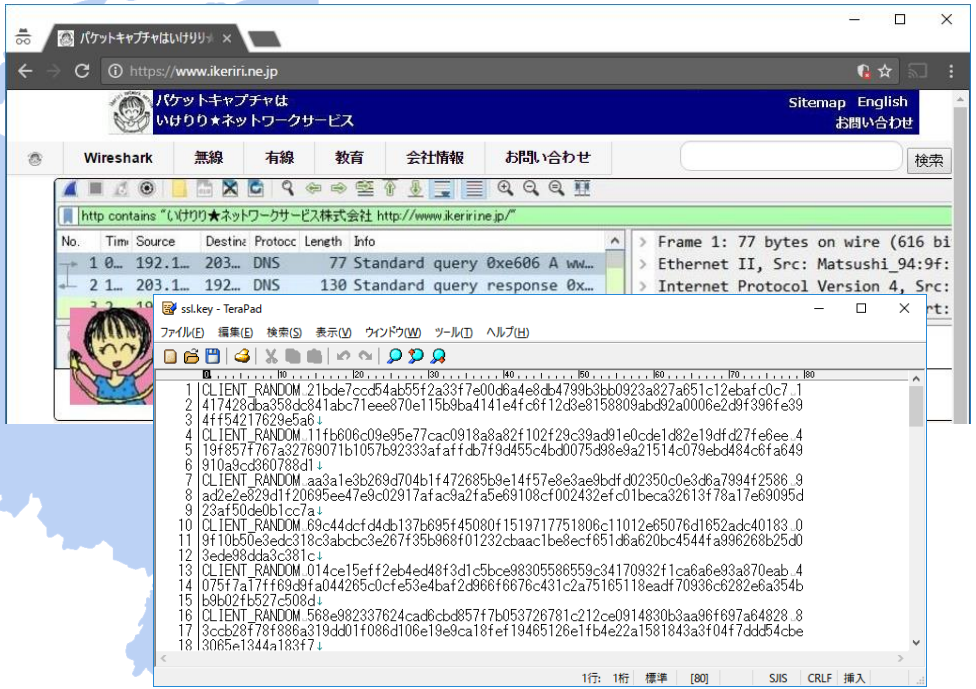
1. Set system environment variable SSLKEYLOGFILE=Path of the premaster secret



3. Decrypting TLS/SSL data without key pair

2. Capture packets using Chrome (https://www.ikeriri.ne.jp/) ssl2.pcapng
   1. Check SSLKEYLOGFILE was generated

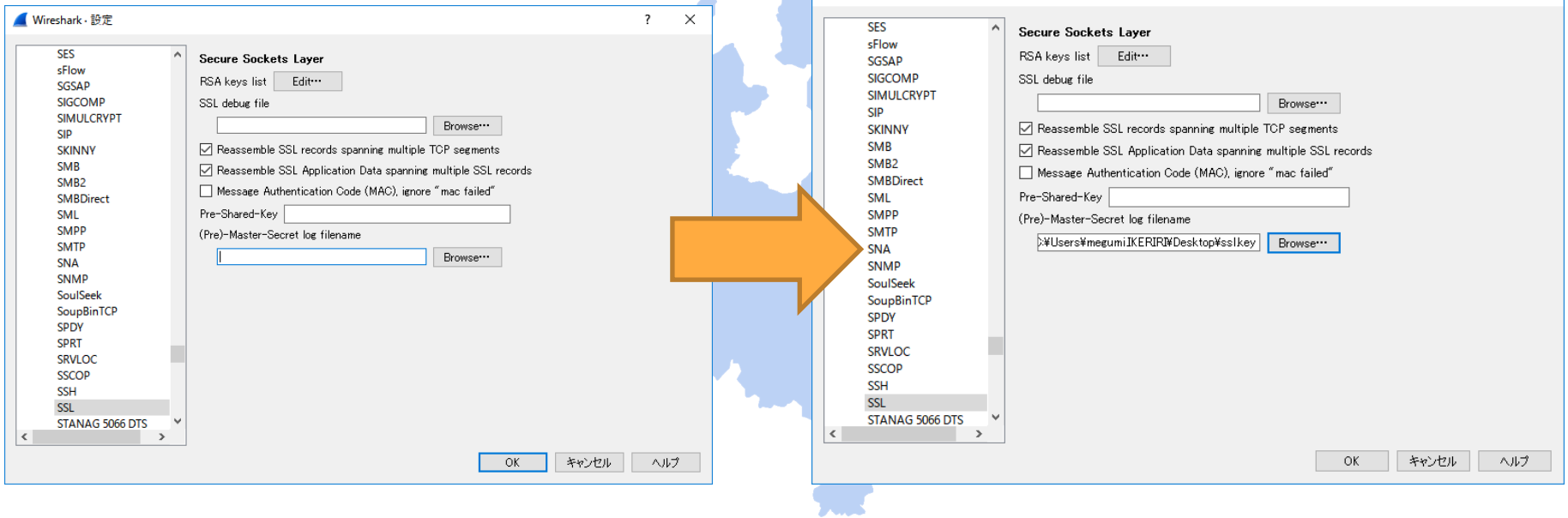




3. Decrypting TLS/SSL data without key pair

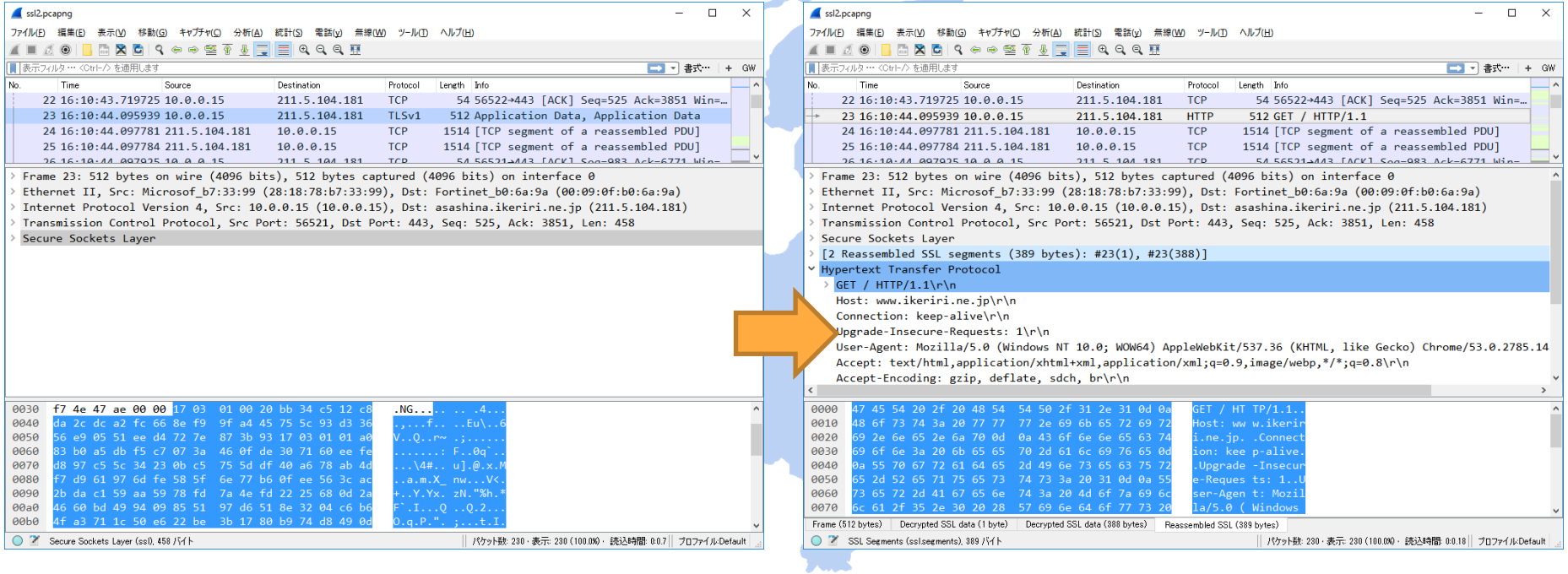
3. Set (Pre)-Master-Secret log filename in SSL preference

ssl.key



3. Decrypting TLS/SSL data without key pair

Check #23,28 in trace file ssl2.pcapng



4.Retrieving Unicode Characters

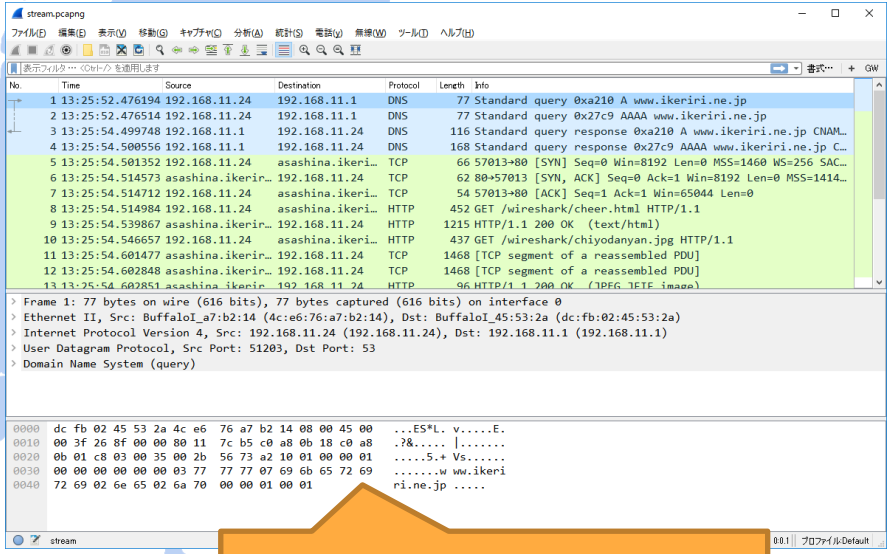
4.Retrieving Unicode Characters

1. Capture packets including Unicode Website (http://www.ikeriri.ne.jp/wireshark/cheer.html)
2. Choose TCP packet, select Follow TCP Stream Byte stream between client and server shows up
3. Select “UTF-8” from list box of “Show and save data as”, we can read Japanese contents !

4.Retrieving Unicode Characters

1. Capture packets including Unicode Website (http://www.ikeriri.ne.jp/wireshark/cheer.html)

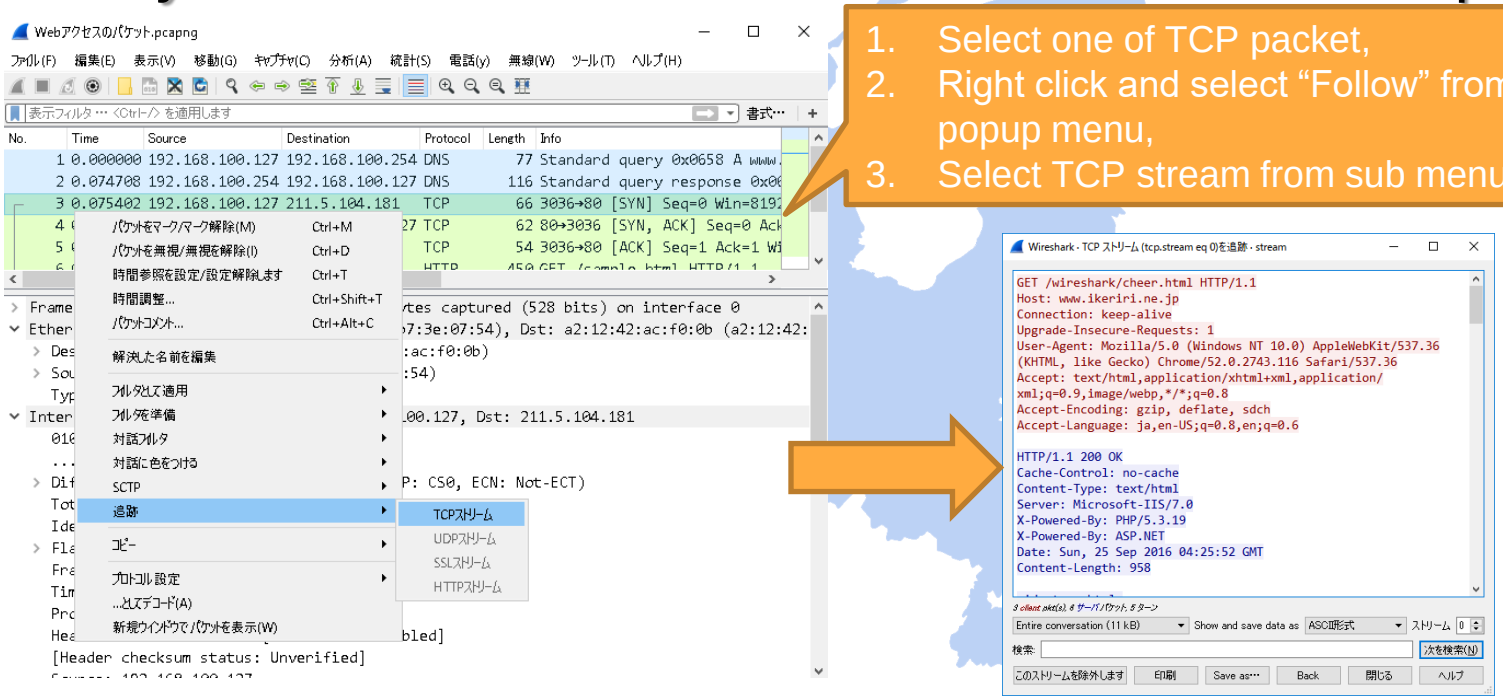
Unicode(Japanese) webpage



stream.pcapng

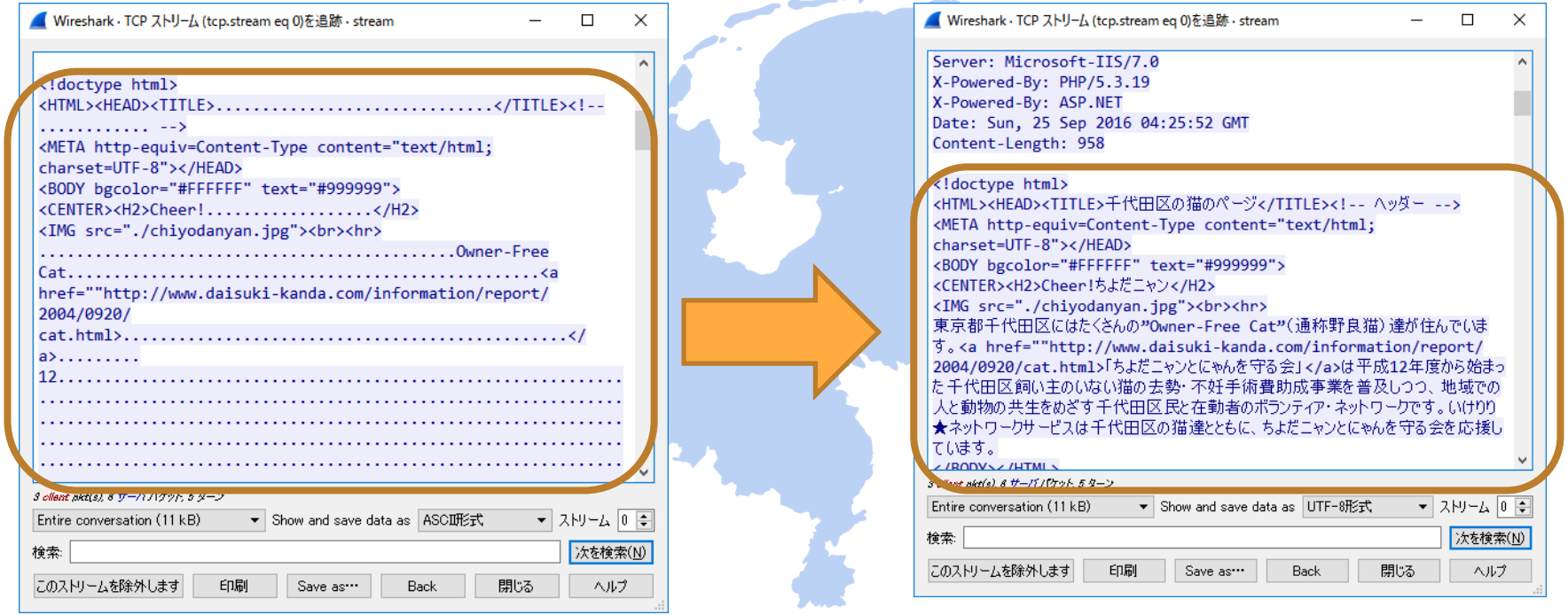
4.Retrieving Unicode Characters

2. Choose TCP packet, select Follow TCP Stream Byte stream between client and server shows up

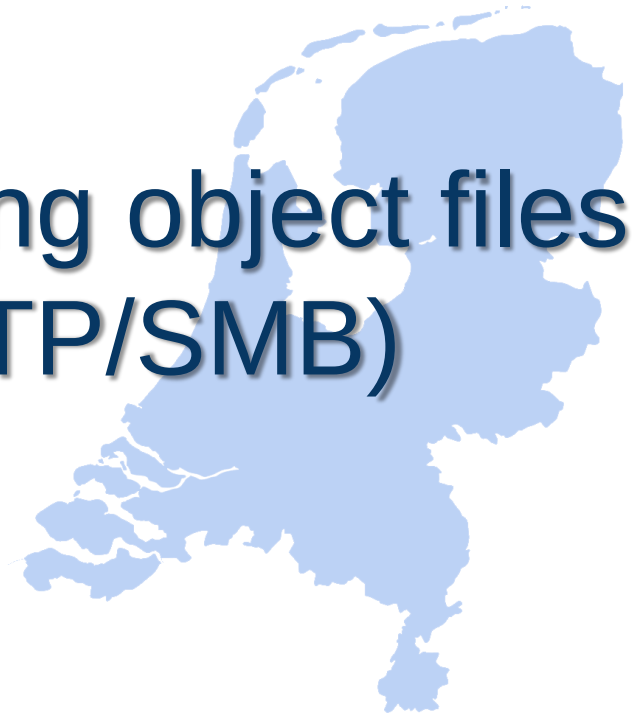


4.Retrieving Unicode Characters

3. Select “UTF-8” from list box of “Show and save data as”, we can read Japanese contents !



5. Retrieving object files (HTTP/TFTP/SMB)



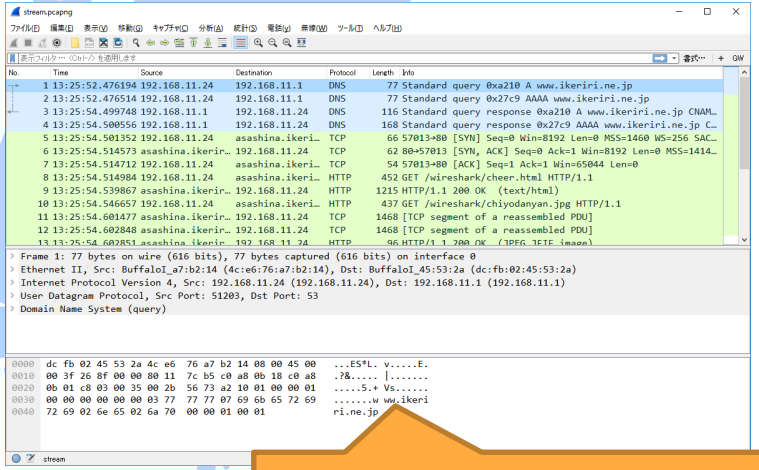
5. Retrieving object files (HTTP/TFTP/SMB)

1. Capture packets including object (HTTP/TFTP/SMB) (http://www.ikeriri.ne.jp/wireshark/cheer.html) note do not use cache mechanizm
2. Choose File > Export Objects > HTTP (also DICOM / SMB / TFTP) and HTTP Object list shows up
3. Select “Save All” into a temporal folder and, we can retrieve CSS, ICO, JPEG, PNG, HTML, ZIP, and more.

5. Retrieving object files (HTTP/TFTP/SMB)

1. Capture packets including object (HTTP/TFTP/SMB) (http://www.ikeriri.ne.jp/wireshark/cheer.html) note do not use cache mechanism

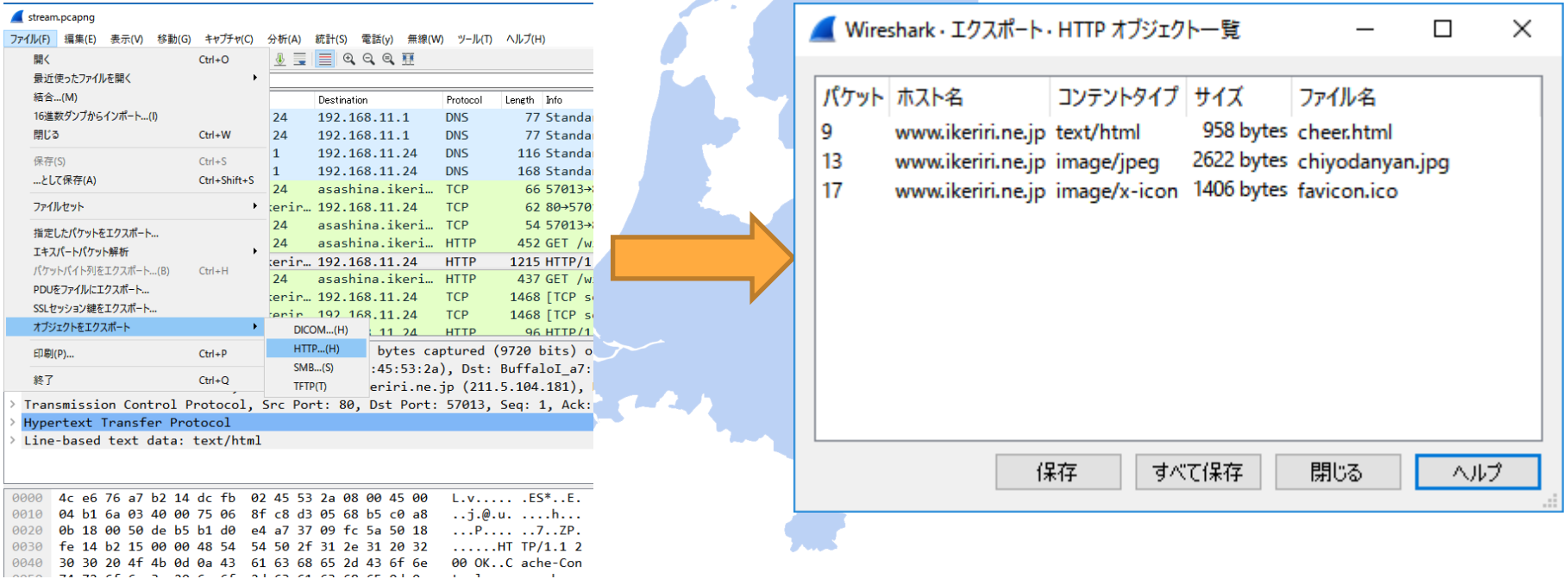
Unicode(Japanese) webpage



stream.pcapng

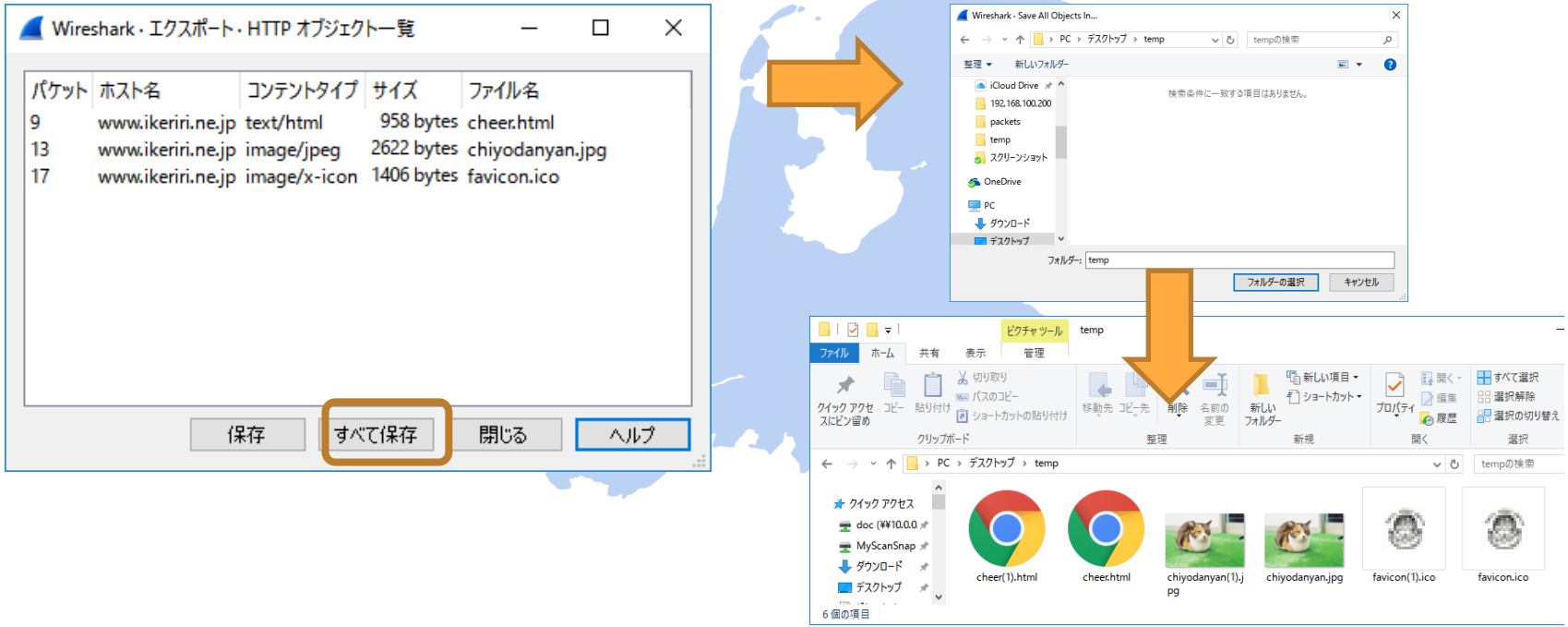
5. Retrieving object files (HTTP/TFTP/SMB)

2. Choose File > Export Objects > HTTP (also DICOM / SMB / TFTP) and HTTP Object list shows up



5. Retrieving object files (HTTP/TFTP/SMB)

3. Select “Save All” into a temporal folder and,
   we can retrieve CSS, ICO, JPEG, PNG, HTML, ZIP, and more.




6. Retrieving values of field

6. Retrieving values of field

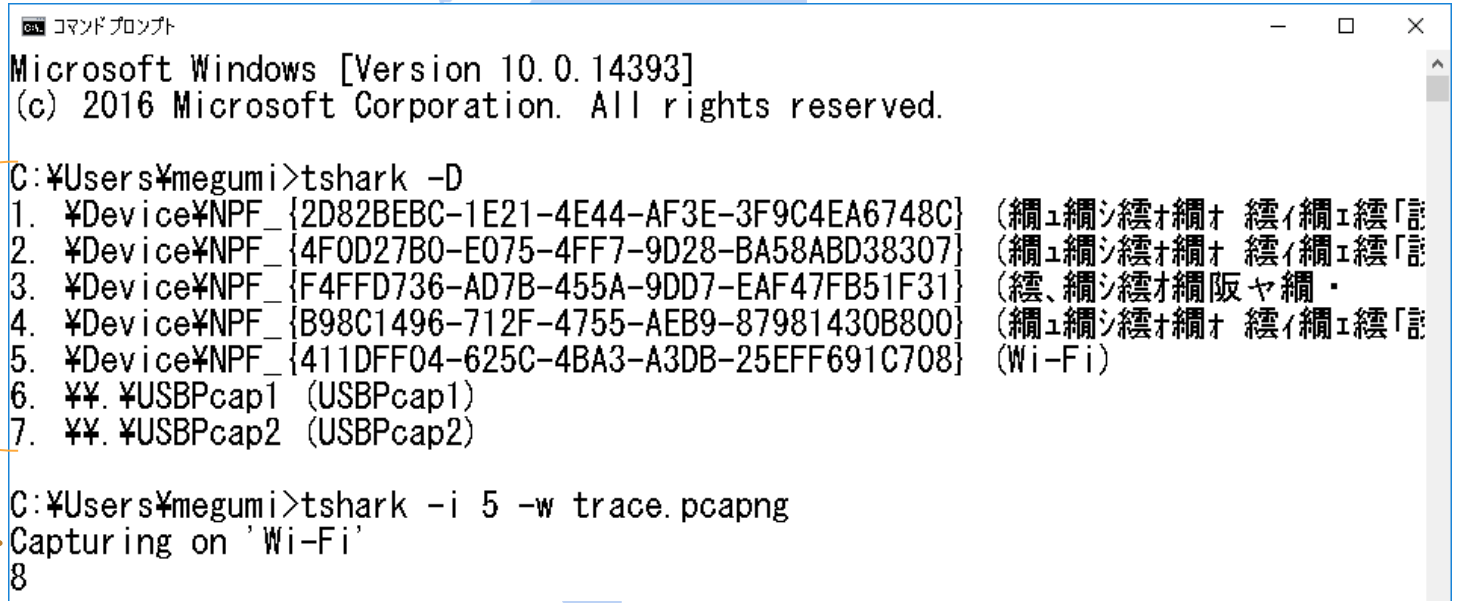
1. If you need data values of field in the trace,
   tshark is the best CLI tool (including Wireshark)
   Use –D option to check interface,
   capture packet with –i interface
   write trace file with –w trace file.
2. tshark has „–T fields“ option to get the value of field
   with „-e“ Display filter (common with -r tracefile)
3. Tshark is more useful connecting another command
   with redirect, pipe (Windows 10 has bash shell)

6. Retrieving values of field

1. If you need data values of field in the trace, tshark is the best CLI tool
   Use –D option to check interface, capture packet with –i interface
   write trace file with –w trace file.

Same order,
Same setting
as default
Wireshark setting,

Set –i interface,
-w trace file
and capture
packets in CLI
(dumpcap is
better)



6. Retrieving values of field

2. tshark has „–T fields“ option to get the value of field with „-e“ Display filter (common with -r tracefile)

コマンド プロンプト

```
C:¥Users¥megumi¥Desktop>tshark -T fields -e ip.src -e ip.dst -r stream.pcapng
192.168.11.24    192.168.11.1
192.168.11.24    192.168.11.1
192.168.11.1     192.168.11.24
192.168.11.1     192.168.11.24
192.168.11.24    211.5.104.181
211.5.104.181    192.168.11.24
192.168.11.24    211.5.104.181
192.168.11.24    211.5.104.181
211.5.104.181    192.168.11.24
192.168.11.24    211.5.104.181
211.5.104.181    192.168.11.24
```

Set Output style As field value

Set –e Display filter string of each fields

Tshark checks Each packet and pick up matched value (note : if there are no matched fields, tshark outputs blank line)

6. Retrieving values of field

3. Tshark is more useful connecting another command with redirect, pipe (Windows 10 has bash shell)

tshark -T fields -e dns.time -r stream.pcapng | find "0" | sort

1:Read stream.pcapng, pick up dns response time (dns.time)
note: Not matched packet returns blank line

2:Use pipe to filter whether there are “0” string
note: time value has “0” (may be ☺))

3:Use pipe to sort ascending.

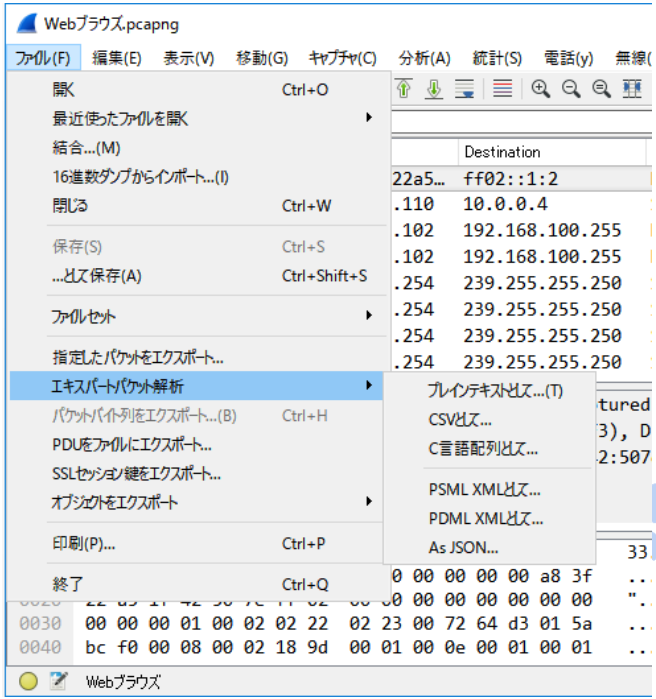
Result 2.023554000
2.024042000

7.Retrieving JSON data

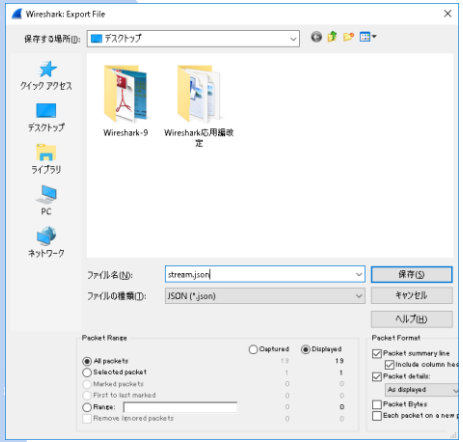
7.Retrieving JSON data

1. JSON (JavaScript Object Notation) is one of open source data format like XML, JSON is commonly used by many web application, data exchange. Wireshark can export packets as JSON.
   You may check JSON file by JSON Viewer.
2. Tshark can also output packets as JSON and EK(ELK) for open source data analysis tool used by Elastic Search, (Logstash,) Kibana.

1. Wireshark can export packets as JSON.

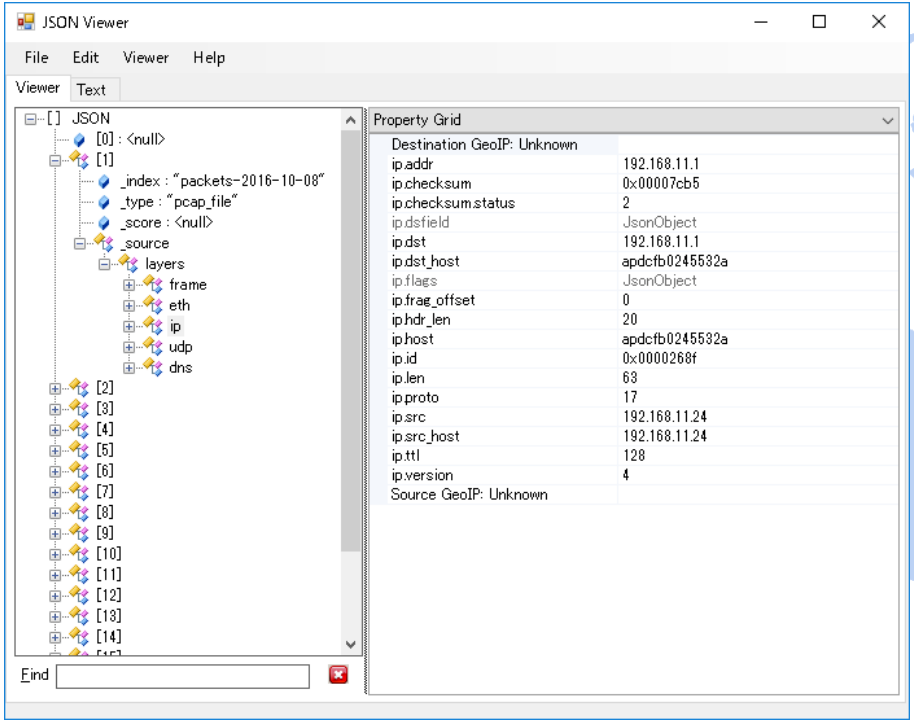


1. Select File > Expert packet analysis > As JSON
2. Set the range of packets, format of file type, and file name as XXXX.json in Export File dialog.

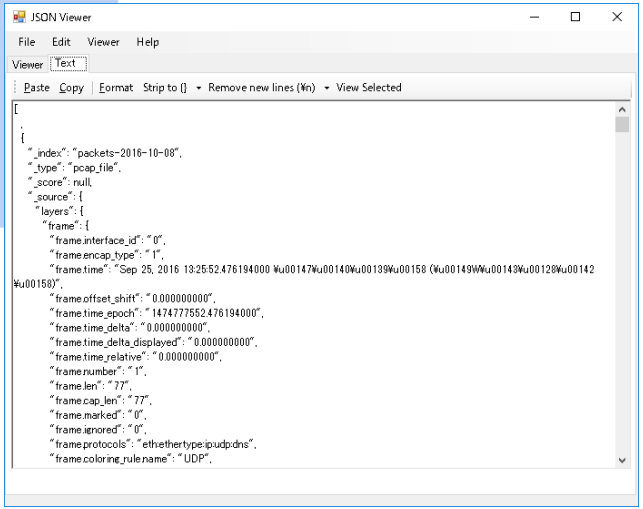


7.Retrieving JSON data

JSON Viewer (http://jsonviewer.codeplex.com/)



„JSON Viewer„ is an open source JSON viewer and editor application on Windows.



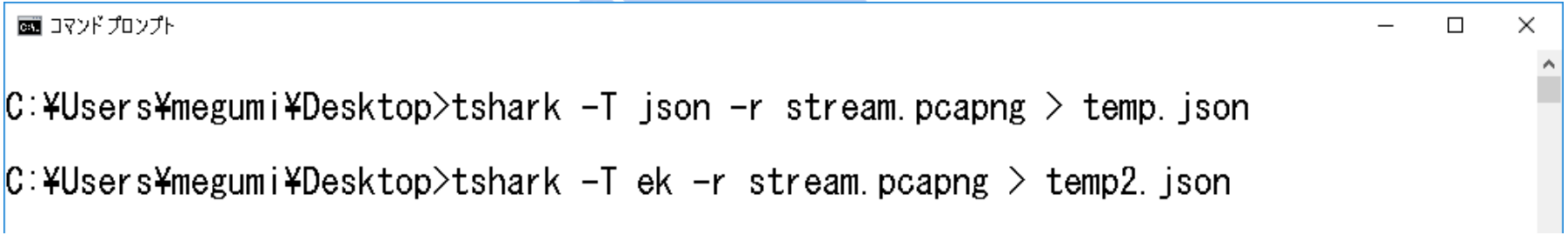
7.Retrieving JSON data

2. Tshark can also output packets as JSON and EK(ELK) for open source data analysis tool used by Elastic Search, (Logstash,) Kibana.

tshark -T (json|ek) –r stream.pcapng > temp.json

1: Read stream.pcapng, convert JSON | EK

2: Redirect output stream to a file named temp.json



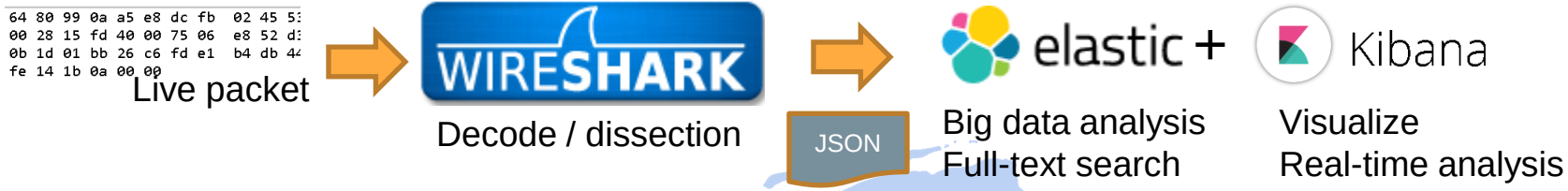
8.Wireshark is the source of big data analysis !

8.Wireshark is the source of big data analysis !

Wireshark is almighty decoder, packet dissection is not only for trace file analysis within Wireshark itself, but also for the source of big data analysis !

We can recode everything in network using Wireshark, and export dissection result as JSON, JSON connect Wireshark with big data analysis.

8.Wireshark is the source of big data analysis !



Elasticsearch is a popular open source full-text search engine based on Apache Lucene, Elasticsearch uses schema-free JSON documents. Kibana is a real-time data visualization platform, a plugin of Elasticsearch. Elasticsearch with Kibana is one of the best open source big data analysis with Wireshark JSON file.

Elasticsearch https://www.elastic.co/products/elasticsearch
Kibana https://www.elastic.co/products/kibana

8.Wireshark is the source of big data analysis !

elastic + Kibana

Set up Elasticsearch with Kibana environment

1. Download JDK, Curl and install, and set system environment variable
2. Download Elasticsearch and start server, check http://localhost:9200
3. Convert packet dissection data into Elasticsearch friendly JSON file
4. Entry packet dissection data (JSON) in Elasticsearch and check data
5. Modify the mapping, re-entry packet dissection data (JSON)
6. Download Kibana and start server, check http://localhost:5601
7. Access Kibana and set index
8. Search packet in full-text, visualize the packet and enjoy big data !

8.Wireshark is the source of big data analysis !

1. Download JDK, Curl and install, and set system environment variable

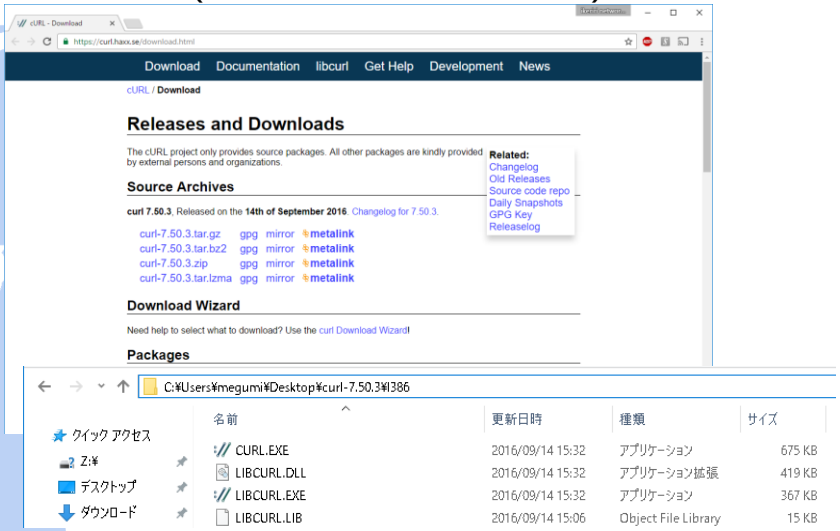
JDK(Java Developer Kit) 8 u101

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download jdk-8u101-windows-x64.exe

Execute and start setup program

Curl (web access command)



https://curl.haxx.se/download.html

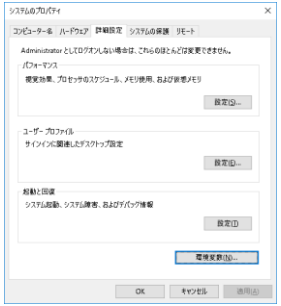
Download curl-7.50.3.cab

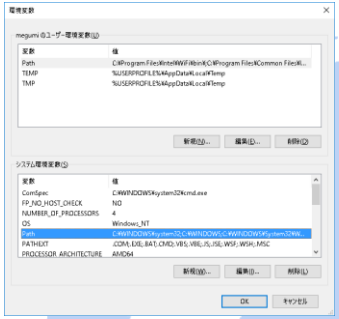
Extract cab file and copy into “Program Files”

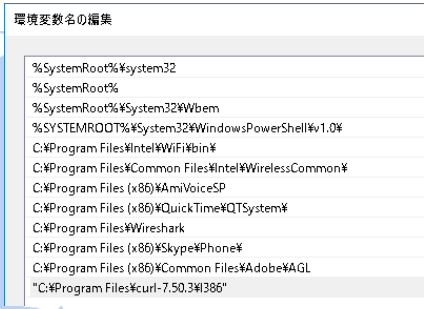
C:¥Program Files¥curl-7.50.3¥i386¥curl.exe

8.Wireshark is the source of big data analysis !

1. Download JDK, Curl and install, and set system environmental variable







Control Panel > System > System setting > detail settings > environmental variable

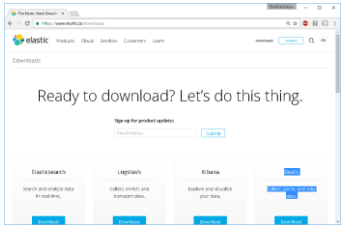
set JAVA_HOME=C:¥Program Files¥Java¥jdk1.8.0_101

set Path=(current path);C:¥Program Files¥Java¥jdk1.8.0_101; C:¥Program Files¥curl-7.50.3¥I386

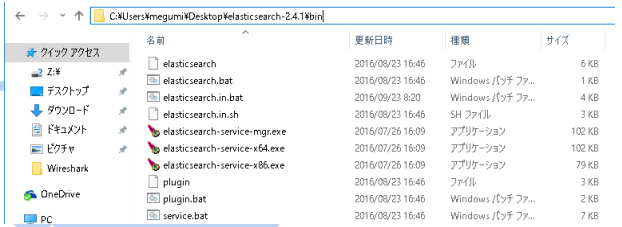
```
C:¥Users¥megumi>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:¥Users¥megumi>curl --version
curl 7.50.3 (i386-pc-win32) libcurl/7.50.3 WinSSL zlib/1.2.8
Protocols: dict file ftp ftps gopher http https imap imaps ldap pop3 pop3s rts
Features: AsynchDNS IPv6 Largefile SSPI Kerberos SPNEGO NTLM SSL libz
```

8.Wireshark is the source of big data analysis !

2. Download Elasticsearch and start server, check http://localhost:9200

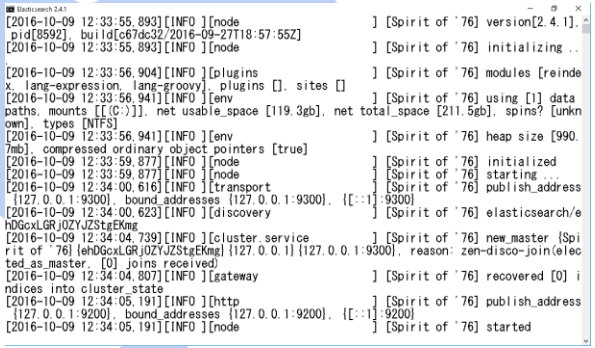


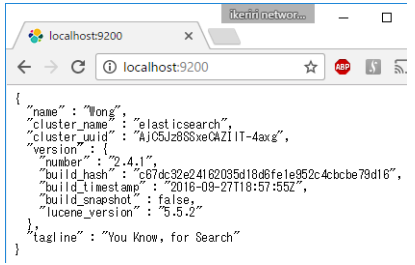
https://www.elastic.co/downloads



C:¥Users¥megumi¥Desktop¥elasticsearch-2.4.1¥bin¥elasticsearch.bat

1.Access https://www.elastic.co/downloads
2.Download elasticsearch-2.4.1.zip
3.Extract zip and open bin folder
4.Execute elasticsearch.bat
5.Check “started” in command prompt
6.Open http://localhost:9200





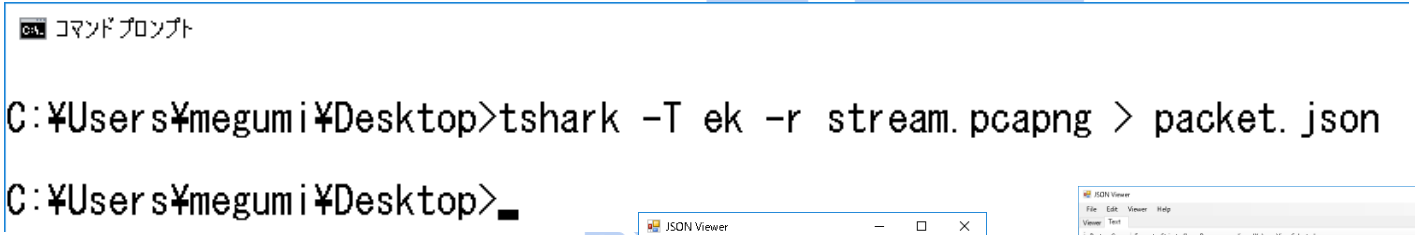
8.Wireshark is the source of big data analysis !

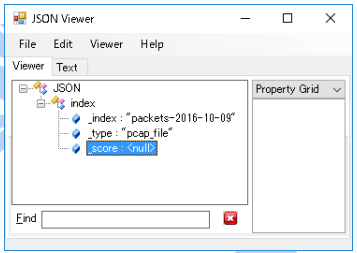
3. Convert packet dissection data into Elasticsearch friendly JSON file

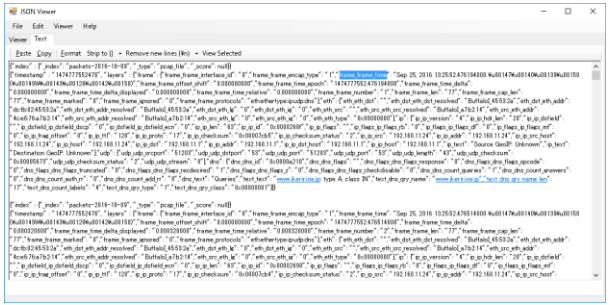
tshark -T ek –r stream.pcapng > packet.json

1: Read stream.pcapng, convert EK (Elasticsearch friendly JSON file)

2: Redirect outut stream to a file named packet.json







8.Wireshark is the source of big data analysis !

4. Entry packet dissection data (JSON) in Elasticsearch and check data

curl –XPOST url @filename : use POST method to send data to server url

curl -XPOST http://localhost:9200/_bulk --data-binary @packet.json

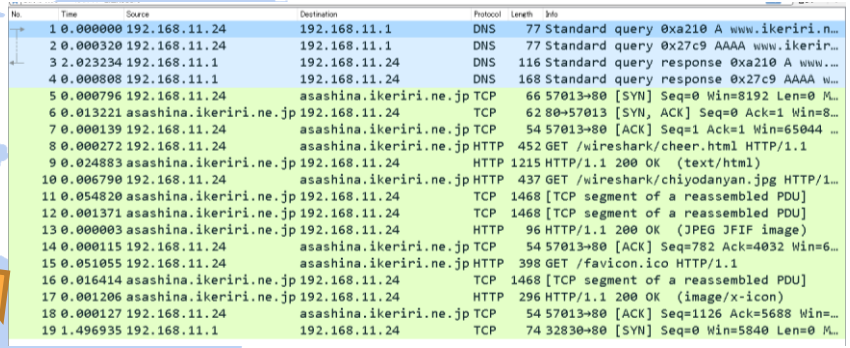
C:¥Users¥megumi¥Desktop>curl -XPOST http://localhost:9200/_bulk --data-binary @packet.json
{"took":5748,"errors":false,"items":[{"create":{"_index":"packets-2016-10-09","_type":"pcap_file","_id":"AVenqQppWAnL3ZFeC-Ci","_version":1,"_shards":{"total":2,"successful":1,"failed":0},"status":201}},...,{"create":{"_index":"packets-2016-10-09","_type":"pcap_file","_id":"AVenqQpqWAnL3ZFeC-C0","_version":1,"_shards":{"total":2,"successful":1,"failed":0},"status":201}}]}

8.Wireshark is the source of big data analysis !

4. Entry packet dissection data (JSON) in Elasticsearch and check data

Access http://localhost:9200/_search?pretty (_search means all index (pretty output) and check data entry


```
{
  "took" : 40,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 19,
    "max_score" : 1.0,
    "hits" : [ {
      "_index" : "packets-2016-10-09",
      "_type" : "pcap_file",
      "_id" : "AVenqQpqWAnL3ZFeC-Ck",
      "_score" : 1.0,
      "_source" : {
        "timestamp" : "1474777554499",
```

4. Entry packet dissection data (JSON) in Elasticsearch and check data


```
{
    "_index" : "packets-2016-10-09",
    "_type" : "pcap_file",
    "_id" : "AVenqQpqWAnL3ZFeC-Ck",
    "_score" : 1.0,
    "_source" : {
      "timestamp" : "1474777554499",
      "layers" : {
        "frame" : {
          "frame_frame_interface_id" : "0",
          "frame_frame_encap_type" : "1",
          "frame_frame_time" : "Sep 25, 2016 13:25:54.499748000
¥u00147¥u00140¥u00139¥u00158
(¥u00149W¥u00143¥u00128¥u00142¥u00158)",
          "frame_frame_offset_shift" : "0.000000000",
          "frame_frame_time_epoch" : "1474777554.499748000",
          "frame_frame_time_delta" : "2.023234000",
          "frame_frame_time_delta_displayed" : "2.023234000",
          "frame_frame_time_relative" : "2.023554000",
          "frame_frame_number" : "3",
          "frame_frame_len" : "116",
          "frame_frame_cap_len" : "116",
          "frame_frame_marked" : "0",
          "frame_frame_ignored" : "0",
          "frame_frame_protocols" : "eth:ethertype:ip:udp:dns"
        },
```

```
{"eth" : {
          "eth_eth_dst" : "",
          "eth_dst_eth_dst_resolved" : "BuffaloI_a7:b2:14",
          "eth_dst_eth_addr" : "4c:e6:76:a7:b2:14",
          "eth_dst_eth_addr_resolved" : "BuffaloI_a7:b2:14",
          "eth_dst_eth_lg" : "0",
          "eth_dst_eth_ig" : "0",
          "eth_eth_src" : "",
          "eth_src_eth_src_resolved" : "BuffaloI_45:53:2a",
          "eth_src_eth_addr" : "dc:fb:02:45:53:2a",
          "eth_src_eth_addr_resolved" : "BuffaloI_45:53:2a",
          "eth_src_eth_lg" : "0",
          "eth_src_eth_ig" : "0",
          "eth_eth_type" : "0x00000800"
        },
        "ip" : {
          "ip_ip_version" : "4",
          "ip_ip_hdr_len" : "20",
          "ip_ip_dsfield" : "",
          "ip_dsfield_ip_dsfield_dscp" : "0",
          "ip_dsfield_ip_dsfield_ecn" : "0",
          "ip_ip_len" : "102",
          "ip_ip_id" : "0x00000000",
```

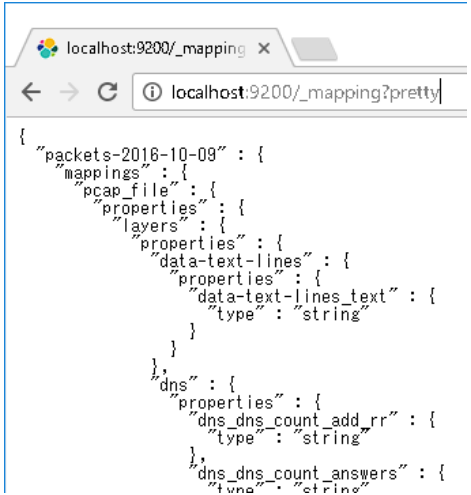
8.Wireshark is the source of big data analysis !

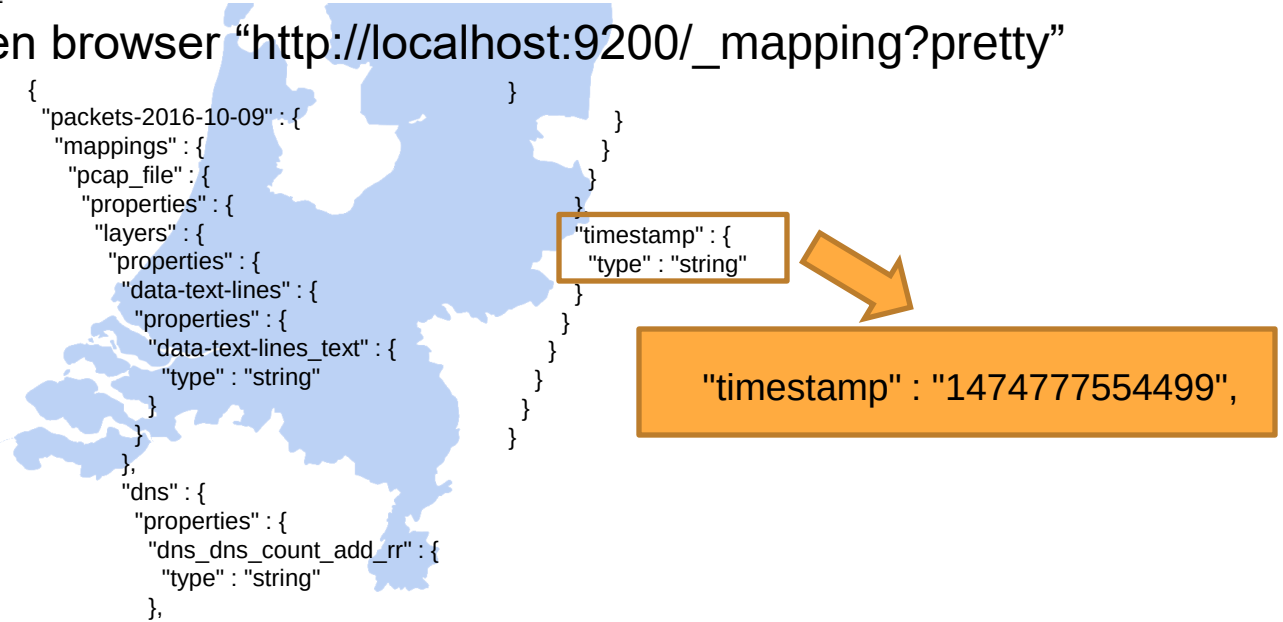
4. Entry packet dissection data (JSON) in Elasticsearch and check data

If you failed to entry data , use curl “curl -XDELETE http://localhost:9200/*”

```
C:¥Users¥megumi¥Desktop>curl -XDELETE http://localhost:9200/*
{"acknowledged":true}
```

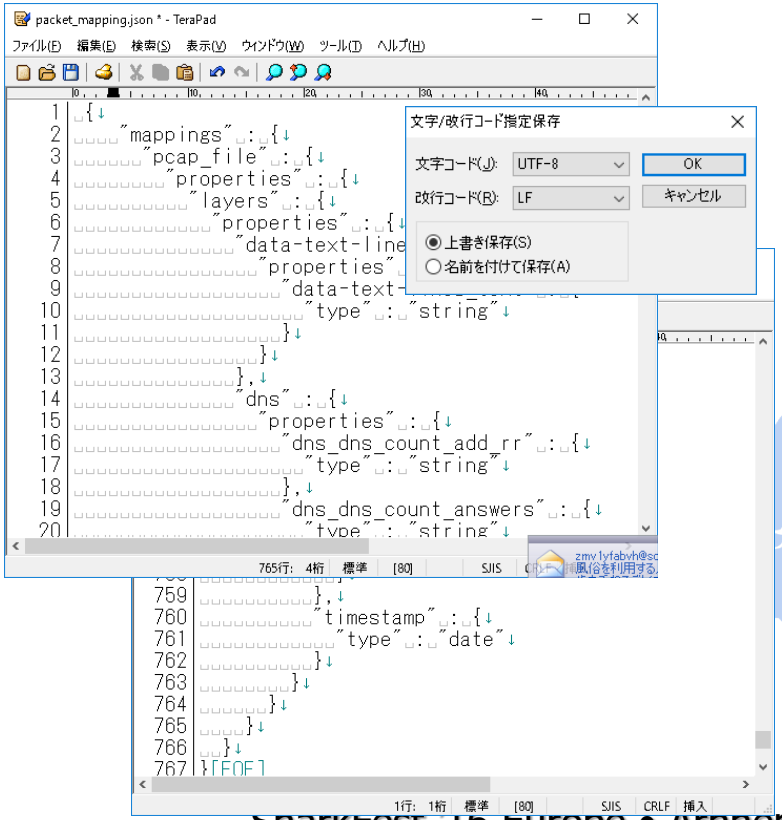
Check data mappings, open browser “http://localhost:9200/_mapping?pretty”





8.Wireshark is the source of big data analysis !

5. Modify the mapping, re-entry packet dissection data (JSON)



http://localhost:9200/_mapping?pretty

Save mapping as “packet_mapping.json”

Delete header **{ "packets-2016-10-09" :**

Modify mapping as

```
"timestamp" : {
    "type" : "date"
```

Check character/return code UTF-8 / LF

Delete “curl -XDELETE http://localhost:9200/*”

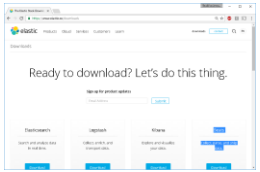
Enter mapping as **“curl -XPOST http://localhost:9200/packets-2016-10-09 --data-binary @packet_mapping.json”**

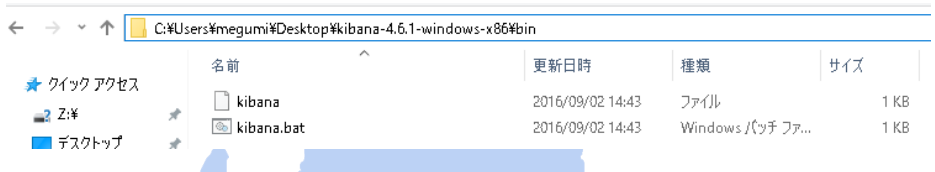
Re-enter original data

curl -XPOST http://localhost:9200/_bulk --data-binary @packet.json

8.Wireshark is the source of big data analysis !

6. Download Kibana and start server, check http://localhost:5601

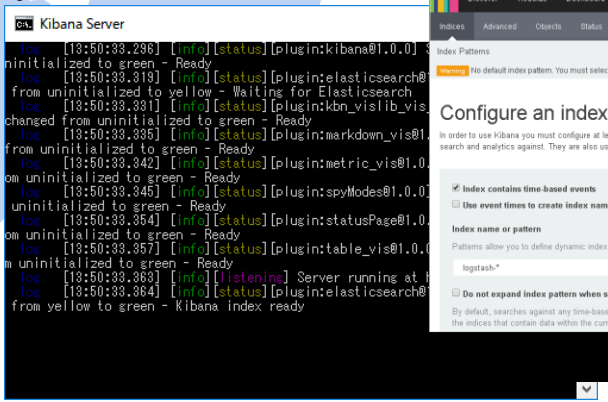


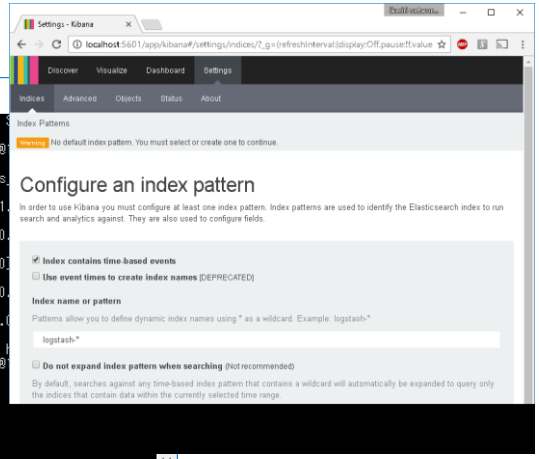


https://www.elastic.co/downloads

C:¥Users¥megumi¥Desktop¥kibana-4.6.1-windows-x86¥bin¥kibana.bat

1.Access https://www.elastic.co/downloads
2.Download kibana-4.6.1-windows-x86.zip
3.Extract zip and open bin folder
4.Execute kibana.bat
5.Check “Kibana index ready”in prompt
6.Open http://localhost:5601

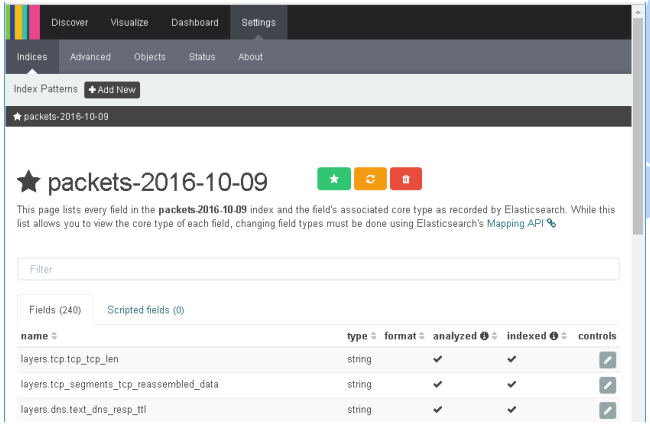


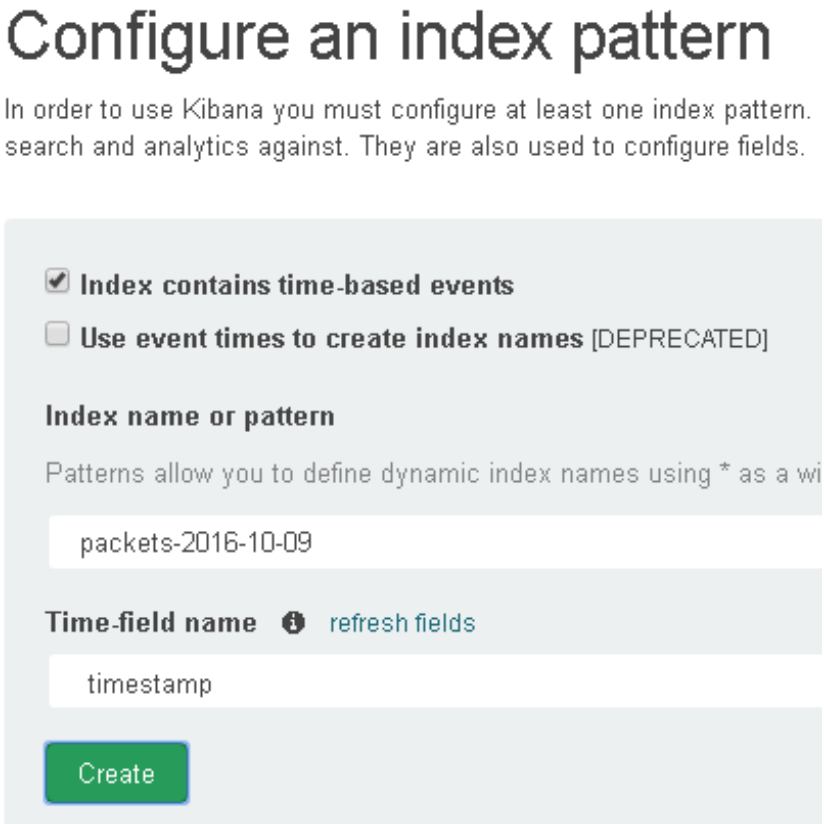


8.Wireshark is the source of big data analysis !

7. Access Kibana and set index

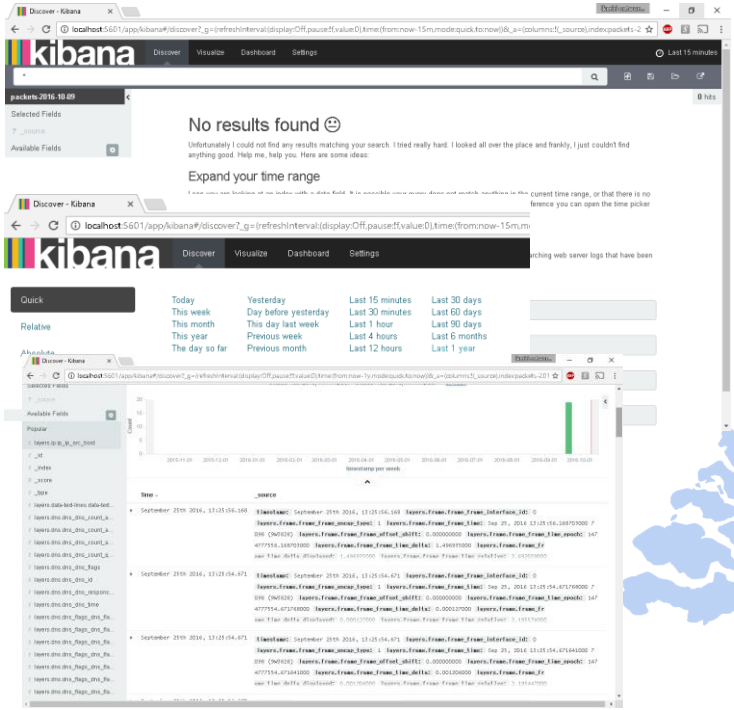
1.Access http://localhost:5601
2.Check “index contains time-based events”
3.Set Index name or pattern as
“packets-2016-10-09” or “packets-*”
4.Set Time-field name as “timestamp”
5.Click “Create” button



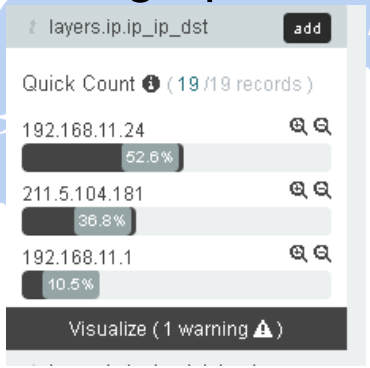


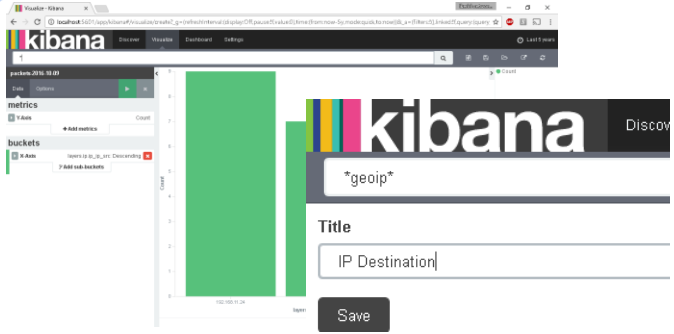
8.Wireshark is the source of big data analysis !

8. Search packet in full-text, visualize the packet and enjoy big data !



1.At First, use time picker to select the time of packets (just use “Last 1 year” is a good way)
2.Check histgram and left pane
3.Select layers.ip_ip_dst in left pane,
click “add” and click “Visualize”,see and save the graph as name “IP ”

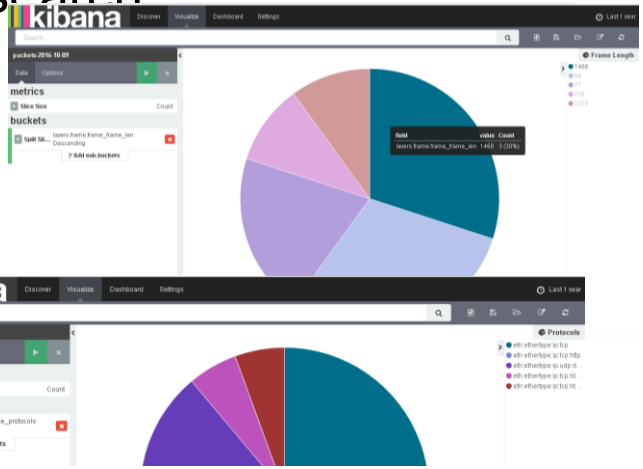




8.Wireshark is the source of big data analysis !

8. Search packet in full-text, visualize the packet and enjoy big data !

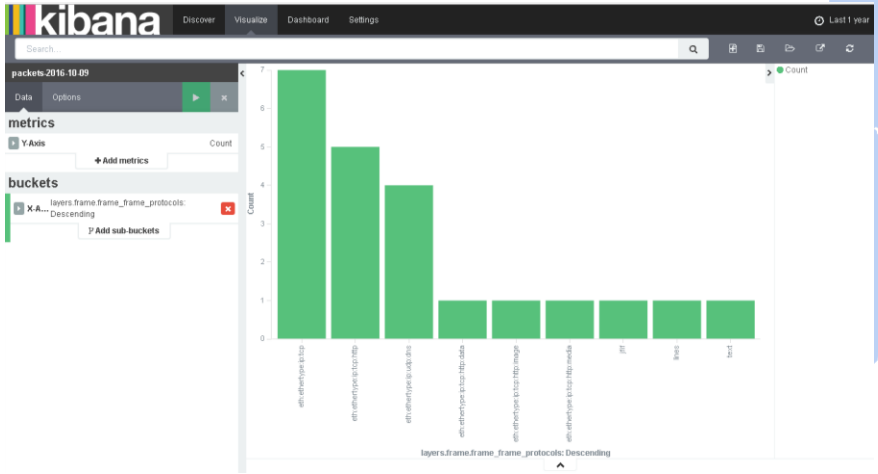
1. Select field ip.ip_ip_src, visualize and save the visualization as “IP Source”
2. Select “Visualize”, “Pie chart”, and “From a new search”
   check “Select buckets type”, click “Split slices”,
   select “Terms” in Aggrigation list box,
   choose field “layers.frame.frame_frame_len”,
   Apply changes, save the visualization
   as “Frame length”
3. Using “layers.frame.frame_protocol” and
   create pie chart, save as “Protocols”
4. Click “Daskboard” and set layout of
   these 4 Visualization



8.Wireshark is the source of big data analysis !

8. Search packet in full-text, visualize the packet and enjoy big data !

Just a few step,
We can create grate visualization
of packets, and enjoy big data !!



THANK YOU VERY MUCH FOR LISTENING

Use Wireshark for everything !

Thank you !
どうもありがとうございます！