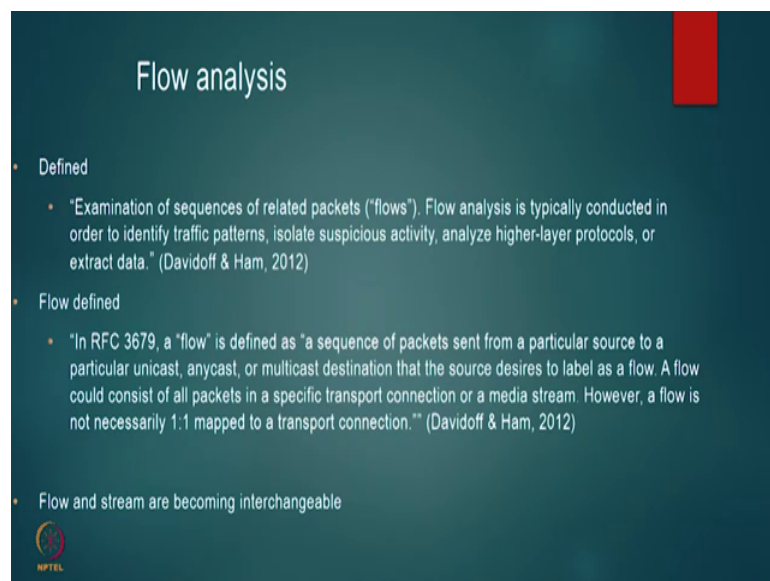


Information security - IV
Prof. M J Shankar Raman
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 40
Flow Analysis
Network Security and Forensics


Until now we have seen how to do protocol analysis and packet analysis. Now what we will try to do is learn about flow analysis and then how to do higher level protocol analysis. There are lots of tools that are available to do higher level protocol analysis. So, I mean if you go to Kali Linux for specific protocols, they are provided analysis. So, once you get the data packets, then you can run these tools to analyze what happens with those protocols.

(Refer Slide Time: 00:49)



Flow analysis

- Defined
 - "Examination of sequences of related packets ("flows"). Flow analysis is typically conducted in order to identify traffic patterns, isolate suspicious activity, analyze higher-layer protocols, or extract data." (Davidoff & Ham, 2012)
- Flow defined
 - "In RFC 3679, a "flow" is defined as "a sequence of packets sent from a particular source to a particular unicast, anycast, or multicast destination that the source desires to label as a flow. A flow could consist of all packets in a specific transport connection or a media stream. However, a flow is not necessarily 1:1 mapped to a transport connection." (Davidoff & Ham, 2012)
- Flow and stream are becoming interchangeable



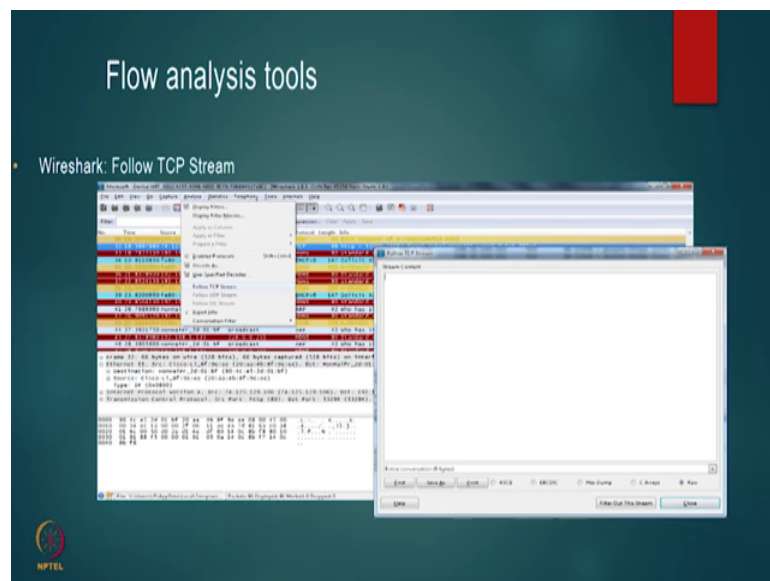
First let us talk about definition of a flow analysis ok. It is an examination of sequence of related packets. So, that is called as a flow ok. Flow analysis you can also call it as sometimes they are call it also a stream analysis. In fact, wire shark uses the word TCP stream. Flow and this is typically conducted in order to identify traffic patterns and then isolate suspicious activity and analyze higher layer protocols or extract data.

So, if a flow is actually defined in RFC 3679 it is a sequence of packets sent from a particular source to a particular unicast anycast or multicast destination that the source

desires to label as a flow. A flow could consist of all packets in a specific transport connection or a media stream. However, a flow is not necessarily one is to on map to transport connection. So, the ideally I mean these are all very formal definitions. So, what we understand is that, we have something known as a session a session is established between two endpoints for some sort of communication. All the packets that ensure that the session is taken care correctly you can call it as a flow ok. For example, I open FTP connection to a particular server.

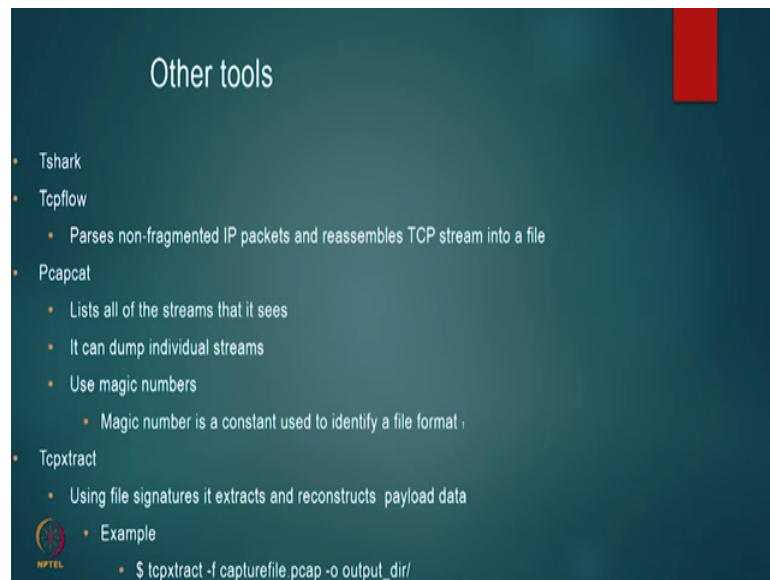
Now, you actually this establishes a session to a particular server. So, all the packets that I use to transfer my file through FTP to other server including the acknowledgements that I receive if there is any usually UDP does not send any acknowledgment, but if there is any then I would like to term that as a flow.

(Refer Slide Time: 02:39)



Now, actually we can use wire shark to do this flow analysis, the easiest way is to just go to analyze and then if you look at follow TCP stream essentially TCP establish some kind of session. So, this can actually go ahead and get what is happens with that particular TCP flow.

(Refer Slide Time: 02:59)



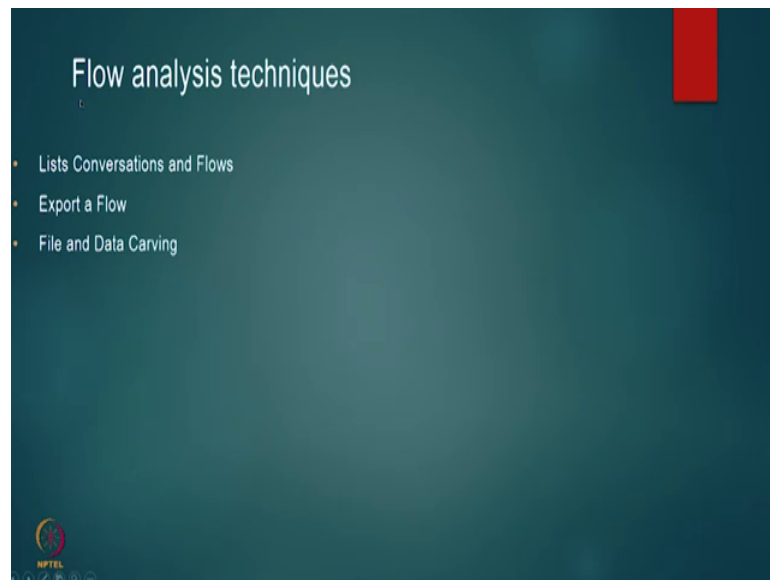
Other tools

- Tshark
- Tcpflow
 - Parses non-fragmented IP packets and reassembles TCP stream into a file
- Pcapcat
 - Lists all of the streams that it sees
 - It can dump individual streams
 - Use magic numbers
 - Magic number is a constant used to identify a file format
- Tcpextract
 - Using file signatures it extracts and reconstructs payload data
 - Example
 - `$ tcpextract -f capturefile.pcap -o output_dir/`

So, it is much easier there are other tools that you can look at the first one as you know is equivalent to wire shark as t shark, then you can have a tool called TCP flow or p cap cat ok. So, a p cap cat the installation is slightly more complicated you or you can use TCP extract and almost all these tools you can just use to identify the more or less when some feature is available in certain tools some is not available.

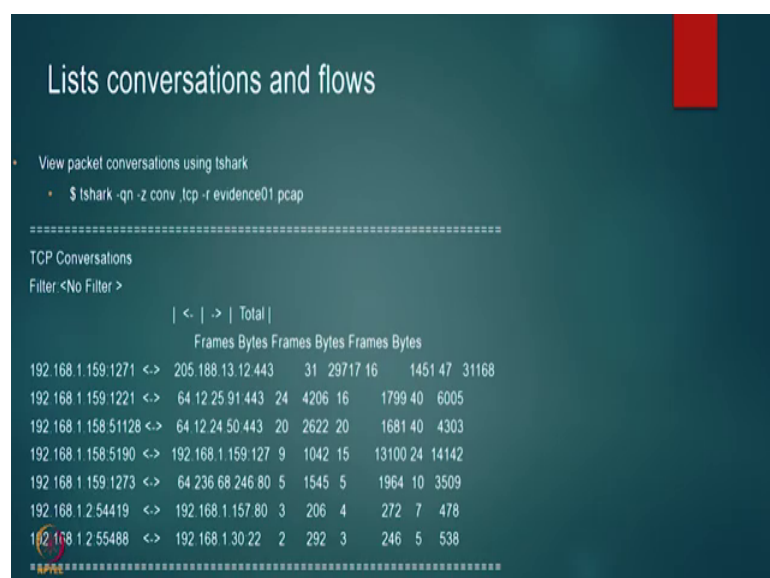
So, you can actually do a comparative study of what t shark has and what TCP flow has what pcp extract has and so on. It is really a good idea to know a lot of these tools. So, that your work jobs we automating these your work becomes much more easier ok. So, how do I how do we do flow analysis.

(Refer Slide Time: 03:48)



One is we just list the conversations and flows and then the particular flow of interest that we have we just export it and then using that we try to identify that that whatever we are exported you try to identify, what is the data that is of relevance to us during network forensics. Do not worry about many of these items being big we will actually do be doing a case study after this module. So, there we will try to correlate what we have learned and then along with the case study. So, you will be much more familiar and after that if you go ahead and revise this, probably you will be able to understand things much better.

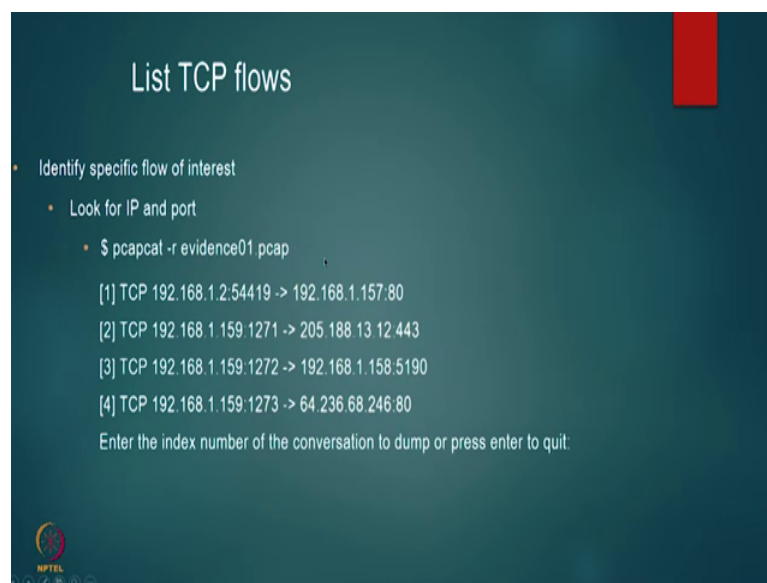
(Refer Slide Time: 04:34)



So, how do I list the conversation flows straightforward we can use t shark and then what we do is using t shark, we just get the conversations that are based on TCP from the evidence file that we can get ok. And once you do that you get something like this it says what is the number of frames and from which connection. So, if you look at this tells you that there is a secure socket connection that has happened here. So, because the port number is 423, then this tells you that there has been a web connection that has happened between these two machines and so on.

So, looking at the port numbers you will be able to identify that, there are a bunch of these conversations that have happened one conversation has happened via the web due to a web server, there is another conversation that happened via web server then there are three conversations that happened via 443, which is SSH. So, in this way and port number 22; so in this way you will be able to there is there something specific about port number 22 ok. So, just take a look at it ok. So, in this way we will able to identify the various conversations and flows that happened.

(Refer Slide Time: 05:42)



Then once we identify. So, what are the flows of interest for example, I want to identify what this person has done after going to the web server.

So, it is http 80. So, I can now take a look at the IP and the port number and then I try to identify what this person has (Refer Time: 06:00) has done this. So, if you use this pcap

cat tool then you will be able to dump all the data that has happened that has got transferred between these two ports and these two machines.

(Refer Slide Time: 06:13)

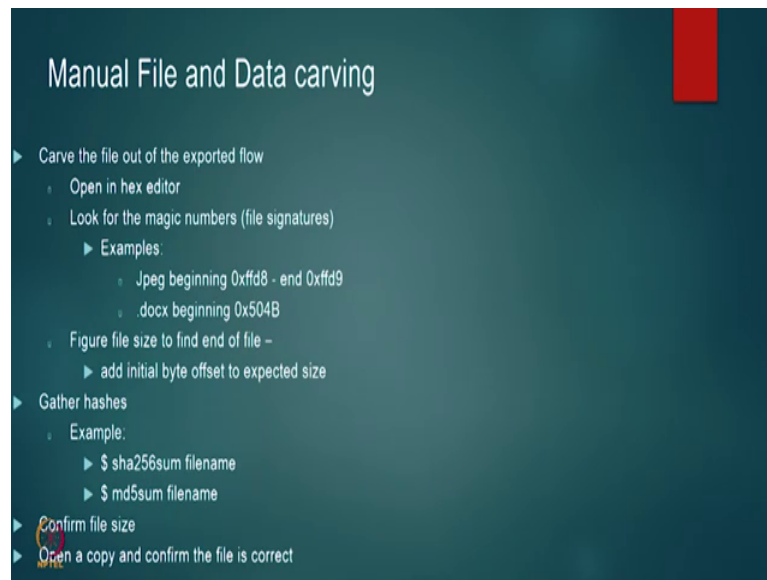
Export a Flow

- Identify the file that most likely contains the evidence for export
 - `$ pcapcat -r evidence01.pcap -w internal -stream dump -f 'host 192.168.1.158 and port 5190'`
[1] TCP 192.168.1.159:1272 -> 192.168.1.158:5190
Enter the index number of the conversation to dump or press enter to quit: 1
Dumping index value 1
 - `$ tcpflow -r evidence01.pcap 'host 192.168.1.158 and port 5190'`
 - Example display:

```
tcpflow [25586]: tcpflow version 0.21 by Jeremy Elson <jelson@circlemud.org >
tcpflow [25586]: looking for handler for data link type 1 for interface evidence01.pcap
tcpflow [25586]: found max FDs to be 16 using OPEN_MAX
tcpflow [25586]: 192.168.001.159.01272 -192.168.001.158.05190: new flow
tcpflow [25586]: 192.168.001.158.05190 -192.168.001.159.01272: new flow
tcpflow [25586]: 192.168.001.158.05190 -192.168.001.159.01272: opening new output file
tcpflow [25586]: 192.168.001.159.01272 -192.168.001.158.05190: opening new output file
```
- Wireshark
 - Click on packet and right-click of "Follow TCP Stream"
 - "Save As" in raw format

Once I dump the data ok. So, if you look at wire shark you can use the follow TCP stream if you were I mean you can use TCP flow also to get all these data. Anyway I mean you stick to one specific tool for the time being and if you find that that tool is not providing you the necessary features then move to some other tool. I mean otherwise there will be using too much of tools and then not much of data we can events from any of these tools ok. Then once I dump the particular data, what I will go and do is now here comes the difficult part, because sometimes you might get incomplete data and you might have to make out evidences based on this incomplete data.

(Refer Slide Time: 07:00)



So, what we usually do is you will actually open a hex editor ok. So, you manipulate the bits and bytes for example, you see this Jpeg always begins with 0 x f f d 8 which is called the signature and ends with 0 x f f d 9. So, sometimes it may not happen that you may not get this 0 x f f d 9, then you might have to put insert it at some place and see whether the image makes some logical sense you know.

So, you might have to insert this 0 x f f at different places and see whether the image makes any logical sense, I mean those kinds of manipulations have to be done and that can be done with one of the hex editors. The other thing and you might have to know what are all the signatures for example, the pdf files carries a particular signature, and this is one of the things that is used for example, there is a command called file in Linux it tells you what type of file it is.

Now it actually does this by looking into the signature of the header, and then trying to tell what type of file it is. So, its similar the same strategy is used here one thing is that the data could be live. So, so what we do is for doc x it is 0 x 5 0 4 b and there are many times where I know the beginning and if I know the size of the file, then I might add up those two and then try to see what is the size of the file the other thing that would be very useful is if I can get the hashes ok. So, for example, m d 5 or a s h a or whatever it is, and then it will be able it will be easy for me to verify. Say for example, tomorrow I recreate

a file and then if I can match the hashes, I mean it is a difficult task, but I am just telling you that these are all some possibilities.

If you can match the hash then I have at least a good guess that this is exactly the file that got transferred. So, in that way you be able to identify how to get the data out of these kinds of bytes ok.

(Refer Slide Time: 08:59)

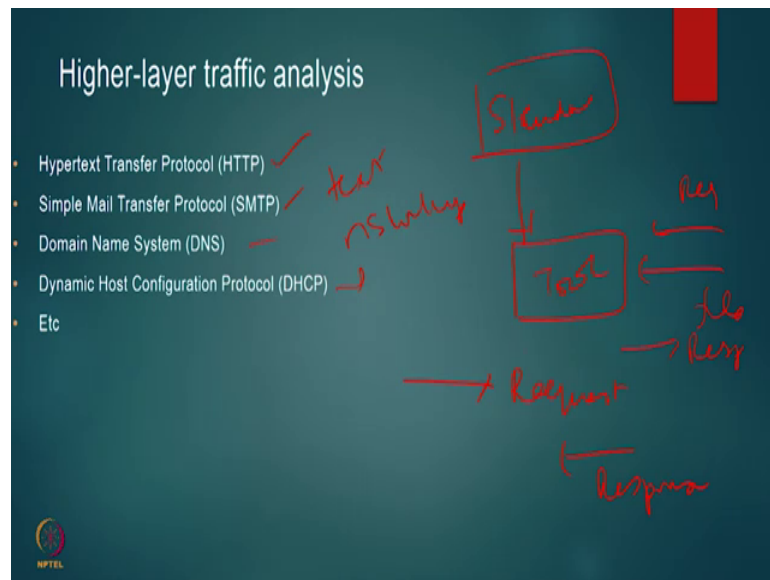


```
Automatic file carving

• $ tcpextract -f evidence01.pcap
...
Found file of type "zip" in session [192.168.1.158:17940 ->
192.168.1.159:63492] , exporting to 00000023.zip
Found file of type "zip" in session [192.168.1.158:17940 ->
192.168.1.159:63492] , exporting to 00000024.zip
Found file of type "zip" in session [192.168.1.158:17940 ->
192.168.1.159:63492] , exporting to 00000025.zip
• $ ls -l
...
-rwx ----- 1 student student 12020 2011 -01 -08 11:22 00000023.zip
-rwx ----- 1 student student 11068 2011 -01 -08 11:22 00000024.zip
-rwx ----- 1 student student 10264 2011 -01 -08 11:22 00000025.zip
```

So finally, ok. So, there are for example, you can also do automatic file carving. So, TCP extract helps you in automatic file carving. For example, in this case you see that there is a zip file. So, what TCP extract tries to do is it tries to take this 179 the data that is transferred between these two ports and then tries to recreate these zip file in this way. And then finally, you can put all these zip files together and then try to recreate the original file. You may not get the whole file, but at least a partial file will also be good enough as a forensic analysis tool. The next part is the higher level traffic analysis.

(Refer Slide Time: 09:34)



Now, in this case you have a lot of tools that are available, now higher level traffic analysis tools at one hand have the protocols. So, what higher level analysis tools will do is ok. So, this is the tool. So, you have the standard which will be the input ok. So, this is the standard is the input, then you have your file the conversation file. So, it tries to find out whether this file follows this standard conversation. So, it is a kind of for example, suppose I the standard says that for every request, I have to get a response message ok. So, what this tool will do is, it will see whether I have got a request packet then for this whether it has got a response packet. So, in that way it is a sort of a comparison that happens between what is the standard and what is there in the in the tool.

. So, in this way you can actually compare HTTP, you can do simple mail transfer see simple mail transfer exactly the protocol is actually text based. So, it will be able to do it. So, domain name system because we know how do you for example, there is a there is a ns lookup ok. So, there is a command called ns lookup, which tries to identify given a domain name what is this IP address or the reverse given the IP address what is the domain name and then this dynamic host configuration protocol, which tries to listen to or try to lease an IP address.

(Refer Slide Time: 11:07)

http

- RFC 2616 defined methods
 - OPTIONS – obtain information about communication
 - GET – retrieve information ID by Uniform Resource Identifier (URI)
 - HEAD – retrieves information without message body
 - POST – send data to URI for processing
 - PUT – upload information to specified URI
 - DELETE – delete resource specified
 - TRACE – echo request message back to client, helpful for debugging
 - CONNECT – reserved

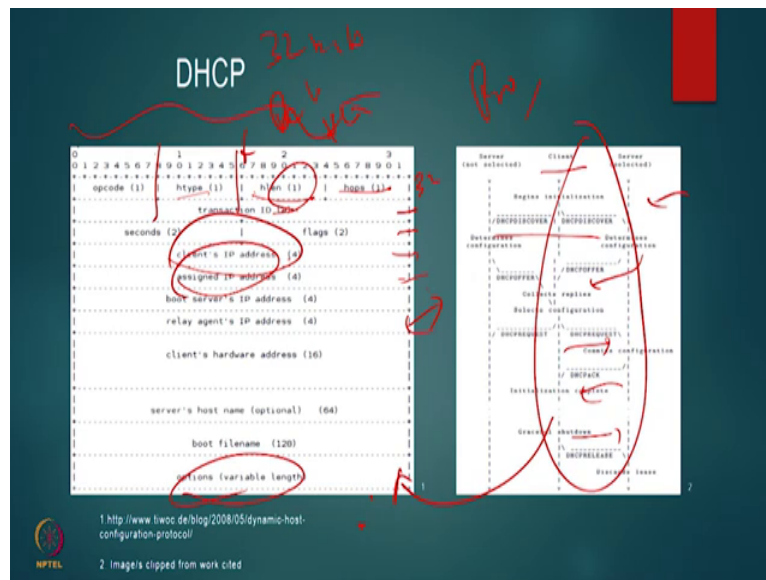
telnet 180
GET 1.1

NPTEL

So, let us look at http for example, this RFC 2616, it defines method for get put post head delete trace etcetera.

Now, the point is if you remember in one of the sessions we did the tel net and then we did it to port 80 ok. So, IP address 480 and then we send this command called get one dot one. So, HTTP actually has this kind of text messages that get transferred and this is defined by this RFC 2616. So, what we could do is we could search for these kind of text messages, and then try to see what is the response that we are getting and then put these sequences together and HTTP tools there are a lot of tools for doing the HTP if you go to Kali Linux it gives you the tools which can put all these flows together.

(Refer Slide Time: 12:00)



The next is DHCP. So, DHCP if you look at this the client actually sends information to who are is the DHCP server.

So, if you remember if you use configure DHCP server ok. So, what happens is that, it tries to do who are all the DHCP servers that are available on the net and then one of the servers will try to offer it the IP address, then this guy contacts the request this person and then this guy acts it and then after the client uses the IP address is just released. So, this is a standard protocol that is defined by the RFC. So, one by do a float trace, I just take these relevant packets and then see whether this protocol is getting satisfied. If I if this protocol is getting satisfied then I can go into the details of each one of this a client address or IP address who is the server etcetera, and then identify the complete conversation that has happened between these two people ok. So, this you first do a protocol analysis, and then I do a packet analysis. I mean I and this could be then if we if we do not get this packets. So, you see this. So, just a brief introduction to what it is.

So, if you look at this, this is divided into four bytes. So, 32 bits and this is for the first 8 bits tells you the op-code the second 8 second bits 8 bits tells you the h type and the header type the header length and then the number of hops etcetera. So, similarly this is this is one word 32 bits. So, this is the second word, third word, fourth word. So, in this protocol the fourth word carries the clients IP address and then the fifth word gives the clients assign what is the IP address that is assign and so on.

So, in this case remember you need header length and because these options are variable length. So, So, using this information I will be able to identify, what all the variable length options that are pass through.

So, in this way I can do first do a protocol analysis and then I go ahead and do a packet analysis and try to identify what has happened ok.

(Refer Slide Time: 13:58)

SMTP

- Important vocabulary
 - Mail User Agent (MUA) – end-users mail client
 - Mail Submission Agent ((MSA) – Local mail submissions
 - Mail Transfer Agent (MTA) – transfers mail between mail servers
 - Mail eXchanger (MX) – accepts incoming messages for a domain
 - Mail Delivery Agent (MDA) – local mail delivery
- Basic commands
 - HELO – opens connection
 - MAIL – identifies return address
 - RCPT – identifies recipient address
 - DATA – message content

Then for SMTP also there are some important vocabulary that we will use, one is the mail user agent and then mail submission agent actually these are formal terms that are used in the RFCs ok. So, essentially if you look at the operation basic commands look at this. Hello, mail received and then data. So, now, this is actually transferred in text format. So, if I get these four keywords in this protocol. So, I will be able to identify say if I get this four keywords, then I will be able to identify what is the exchange that has happened between my client mail client and the mail server; a formally to say mail user agent and the mail exchange servers ok.

So, in this way I will be able to identify what is the protocol higher level protocol, that has worked out.

(Refer Slide Time: 14:52)

DNS

- Query-response protocol
 - Client question = single UDP packet
 - Server response = single UDP packet

The diagram illustrates the structure of a DNS header. It is a 12-byte structure. The fields are: Identification (2 bytes), Opcode (1 byte), DNS Flags (2 bytes), RCode (1 byte), Total Questions (2 bytes), Total Answers (2 bytes), Total Authority Resource Records (2 bytes), and Total Additional Resource Records (2 bytes). The diagram also shows the bit-level structure of the DNS Flags field, which includes bits for QR, AA, TC, RD, RA, Z, and AD.

DNS Header Format
Copyright © 1997, RFC 1035 and RFC 1034

NPTEL | http://www.troyesup.com/headers/DNS_Header.png

And how we can extract information from each of the packets that are there in the higher level a similar stuff can be done with DNS ok. So, in this case what we are trying to do is we are trying to identify the IP address of a particular machine like say www dot Google dot com ok. So, Google this is a DNS address domain name system ok. So, in this case what we do is we sent by a UDP packet to port number 53 to any DNS server, and then the DNS server actually sends you the it follows different types of records if you go into the record into the protocol it talks about double a record triple a record and so on and then sends a response.

So, if you remember. So, this actually it forms a hierarchical kind of system. So, if one server does not have it is the answer for a particular query, DNS query then it passes on to the top of its hierarchy and so on ok.

(Refer Slide Time: 15:55)

Higher-layer analysis tools

- Ofcat
 - Input = reassembled single flow of transport layer payload (ex: tcpflow or pcapcat)
 - Output = protocol summary of all OFT activity and any recovered files transferred
 - <http://blog.kiddalund.net/dw/ofcat>
- Smtpdump

```
# smpdump
smpdump version 0.1
Copyright (C) 2009 FRANK CHERICHOY
smpdump comes with ABSOLUTELY NO WARRANTY;
This is free software, and you are welcome
to redistribute it under certain conditions.
(GPL v3)

Usage: smpdump [Options] -r <pcap_file>
-a, --auth                Display SMTP auth informations only
      LOGIN method)
-e, --info                Display E-mail informations
-b, --brief               Display minime e-mail informations
-m, --extract             Extract e-mail attachments
-M, --MD5                Display extracted attachment MD5 Hash
-s, --save                Save raw e-mail to file
-f, --flow-index <index> Filters only given index flow
-r, --read <pcap_file>  Read the given pcap file (REQUIRED)
-v, --version            Display version information
-h, --help               Display this screen
```

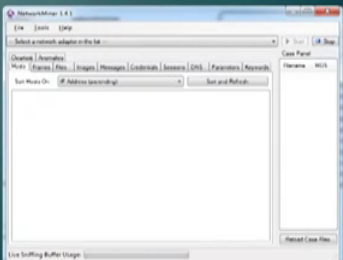
HPTEL2 Images clipped from work cited

Now there are tools like SMTP dump ok. So, these tools actually if you look at this they can actually dump whatever is the SMTP messages that are there ok. So, here is another tool called ofcat ok. So, this tool can actually it takes a reassemble single flow of transport layer payload like TCP flow or a pcapcat and protocol summary of all of activity or any required files that are transferred ok. So, it can do all these kind of now some of these tools you might have to write some of these tools are already existing for example, SMTP dump is there with Kali Linux.

(Refer Slide Time: 16:34)

Higher-layer analysis tools

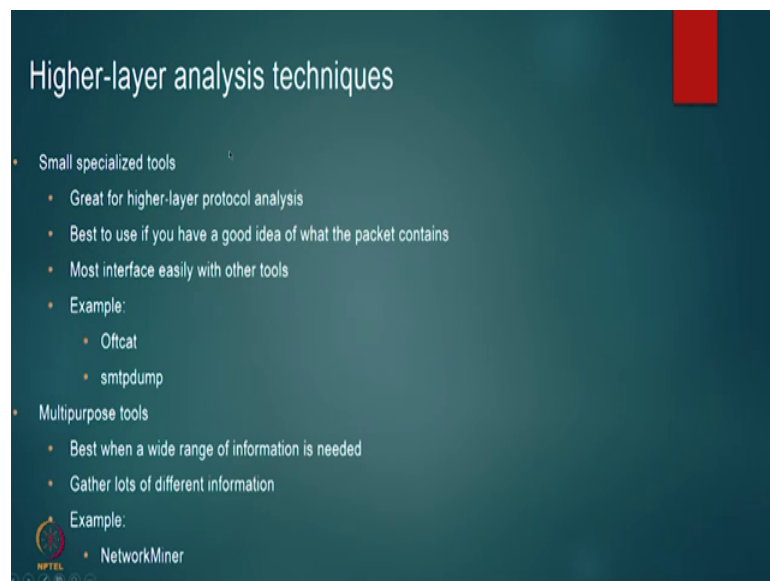
- Findsmtpinfo.py
 - Input = pcap file
 - Output = extracted authentication data, credentials, mail header info, attachments, MD5 sum and produces a report
 - http://forensicscontest.com/contest02/Finalists/Jeremy_Rossi/findsmtpinfo.py
- NetworkMiner
 - Multipurpose traffic analyzer



So, you should be able to make use of these tools to do some analysis. So, there are certain rules I mean if this is a very interesting website called forensic contest dot com. Actually they are dumped some pcap files and then ask you to do some kind of forensic analysis you would suggest that you participate in some of these activities, I mean this is one such we are not advising this website, but it is one such website, where we can go ahead and work on their problems, because by solving some of these problems actually I mean you make you can even write your own software to do some forensic analysis. So, here is one that is written by this person. So, if you look at this you input a pcap file and then output is authenticated data credential mail header info attachment a md 5 sum and then it produces the report.

Similarly, network miner can also be used ok.

(Refer Slide Time: 17:30)



So, the advantage of higher layer techniques is that you can you are (Refer Time: 17:37) to write some small specialized tools and since its higher level protocol analysis actually easy to I mean not easy I would say it is an interesting work to do this kind of analyzers and these can be very successful only if you have an idea of what the packet contains.

So, as we discussed earlier some protocols are not open therefore, you might have to identify what the protocol is, and then write these kinds of tools that makes it very very challenging and so, these are you could also write a kind of multipurpose tools ok. So, for example, a network miner is one tool which actually get gets a lot of information

similar to say wire shark also gives you a lot of information anyway. So, so. So, you should be able to write these kind of higher level protocol tools, that will help you do forensics much faster ok.

So, this is a good reference book which we have been using for this course, and what we will do in the next section is fine we have talked a lot about flow analysis, we have talked about lot about protocol analysis, we have talked a lot about packet analysis. What we will do in the next session is that, we will go ahead and get one evidence file and then work through this whole process, and see what is the problem it is a kind of a case study, what is the problem that was reported and how one is able to identify and solve a problem ok.

Thank you very much.