

# APNIC Training

## Network Analysis and Forensics

In conjunction with

**PACNOG4**

Port Vila, Vanuatu

2-4 July 2008

## Acknowledgements

- Some content in this module is based on material developed and provided by Team Cymru  
[www.cymru.com](http://www.cymru.com)
- APNIC acknowledges their contribution and support with appreciation and thanks, and particularly recognises the collaboration with Ryan Connolly of Team Cymru
- Some material is also sourced from lecture material from the QUT Internetworking course (ITB524)

## Acknowledgement

Material in this module was also sourced from the following publication:

**Guide to Integrating Forensic Techniques into Incident Response: Recommendations of the National Institute of Standards and Technology**

*Karen Kent, Suzanne Chevalier, Tim Grance and Hung Dang, published by National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Commerce*

## Note

- Certain commercial organisations and their products and services are mentioned in this module. However this does not imply recommendation or endorsement by APNIC



# Network Protocol Analysis: Understanding TCP/IP

**Cecil Goldstein**

[cecil@apnic.net](mailto:cecil@apnic.net)

## Outline:

This discussion will look at the nature of network data, and how it can be understood and analysed.

In order to achieve this, the function, structure and operation of the protocols governing data communications must be clearly understood

This tutorial assumes no prior knowledge and aims at providing a basic overview of the structure of the TCP/IP protocols

The use of a packet analyser (Wireshark) will be demonstrated as a tool for monitoring, understanding and analysing network data

## Topics

### Data on the Internet

- What is data
- What is a protocol
- TCP/IP Overview – (IP, TCP)
- “Seeing” network traffic – using a packet analyser
- Wireshark overview, operation, layout, features
- Using Wireshark to understand and analyse network traffic

## Data and Protocols

- Data is transmitted over a physical network as a sequence of binary digits (bits - 0s and 1s).
- The "sending" process involves the source device generating a pattern of signals (voltages, light patterns, wavelengths).
- The pattern of signals generated represents the sequence of bits making up the data.
- These signals can be "read" by any device attached to the same physical network.
- “Reading” means identifying the signals to receive the same pattern of bits as generated by the sender.




## Protocols

- All data is transmitted in the same way irrespective of what the data refers to, whether it is clear or encrypted.
- The data communication protocols define the structure or pattern for the data transferred – this gives it its meaning.
- The Protocols define
  - *functions* or *processes* that need to be carried out in order to implement the data exchange and the
  - *information* required by these processes in order for them to accomplish this


## Modularity

- The processes are *modularised* into *layers*, with each layer responsible for component in the overall data communication process
- For each layer the protocols then stipulate :-
  - The specific actions to be carried out at that layer.
  - the information required or generated at that layer and the format of that information (Headers)
  - The interactions between the layers (how and which information is passed between them)
- The action of all the layers together constitutes the data communication

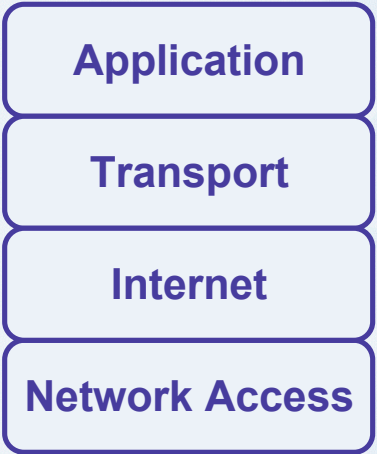
APNICAsia Pacific Network Information Centre

## Analysing data - Forensics

- By understanding the structure of the protocol headers at each layer, and the meaning of the values carried in the header fields, it is possible to understand the behaviour of the network traffic

APNICAsia Pacific Network Information Centre

## The Four Layers of TCP/IP



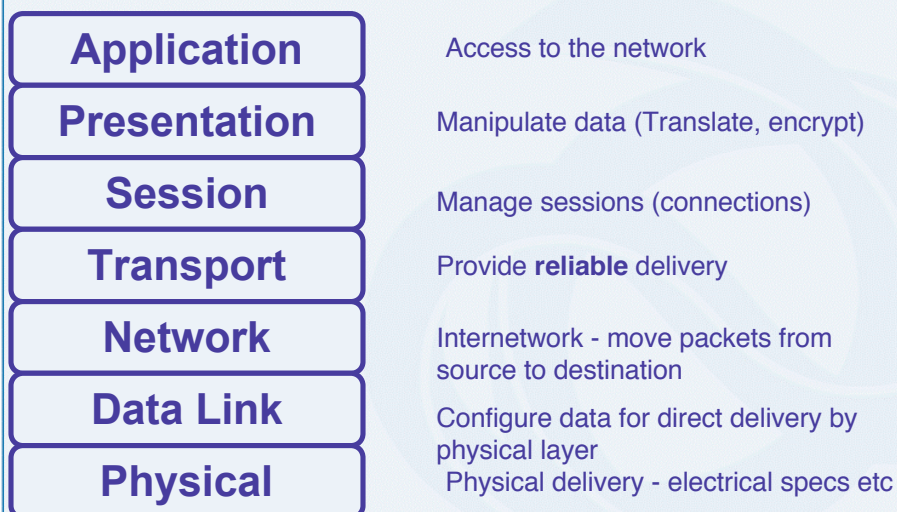
```
graph TD; A[Application] --- B[Transport]; B --- C[Internet]; C --- D[Network Access];
```




## Protocol Models

- In the late 1970s the ISO (International Standards Organisation) introduced a model defining the functions for data communications between two computers in a **7 layer model** - The OSI (Open System Interconnection) Model
- Not a protocol but a framework intended to facilitate the design of protocols for inter-computer communication.
- Defines the processes required at each of the modularised layers
- OSI is “protocol independent”

## Protocol Models The OSI Model






Asia Pacific Network Information Centre

APNIC

## The Four Layers of TCP/IP and the OSI Model

- TCP/IP was created before the OSI model
- It is a layered protocol implementation
- Its layers do not match the OSI model exactly, but the processes defined in the OSI model are contained in the TCP/IP layers



Asia Pacific Network Information Centre

APNIC

## The OSI Model and TCP

Application	Application (HTTP, FTP, SMTP, TELNET )
Presentation	
Session	
Transport	Transport (TCP)
Network	Internet (IP)
Data Link	Network Access
Physical	





## Packet Switching

- Revolutionary “invention”: a concept developed (separately) by Paul Baran (US) and Donald Davies (in the UK).
- Data broken up and sent in independent “packets”
- It is a “connectionless”, multiplexing environment
- Each packet contains all the information needed to get it to its destination and for the processing requirements to achieve this as defined by the protocols.
- At the destination, packets can be put together again to make up original data stream (connection “orientation”)



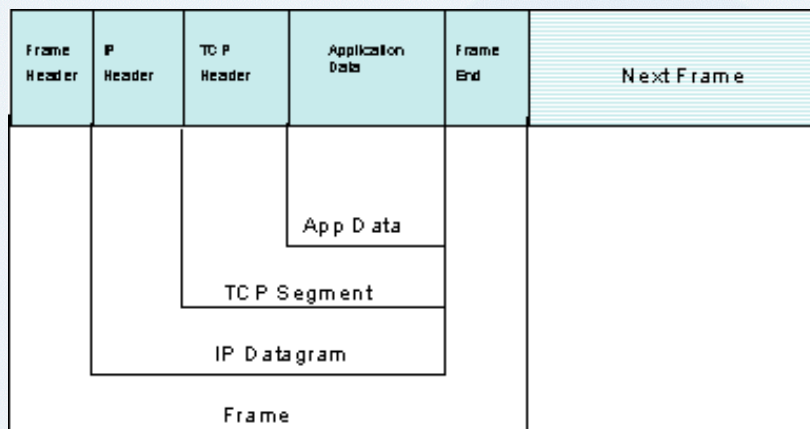
## Packets

- A packet then contains a set of data made of the various headers from each layer including the data generated by the application layer.
- The packet is “built” during a sending process when each layer determines the information needed for its tasks, and adds this header information
- The layer will then take this information, with any other data it might have received from a higher layer, and pass it as one set of data to a lower layer.
- This process is then repeated and is called *encapsulation*


## Data Analysis

- Analysing network traffic is then essentially:
  - Understanding the structure and meaning of protocol headers
  - Understanding what occurs at each stage of the data communication process.
  - Being able to “decapsulate” a packet and identify the relevant headers
  - Knowing what behaviour is expected at each point in the data transfer.
  - Being able to recognise when this behaviour is unusual
  - Being able to identify what header information might be inconsistent and could be causing this behaviour to occur

## Encapsulation








APNIC  
Asia Pacific Network Information Centre

## Data is Data!

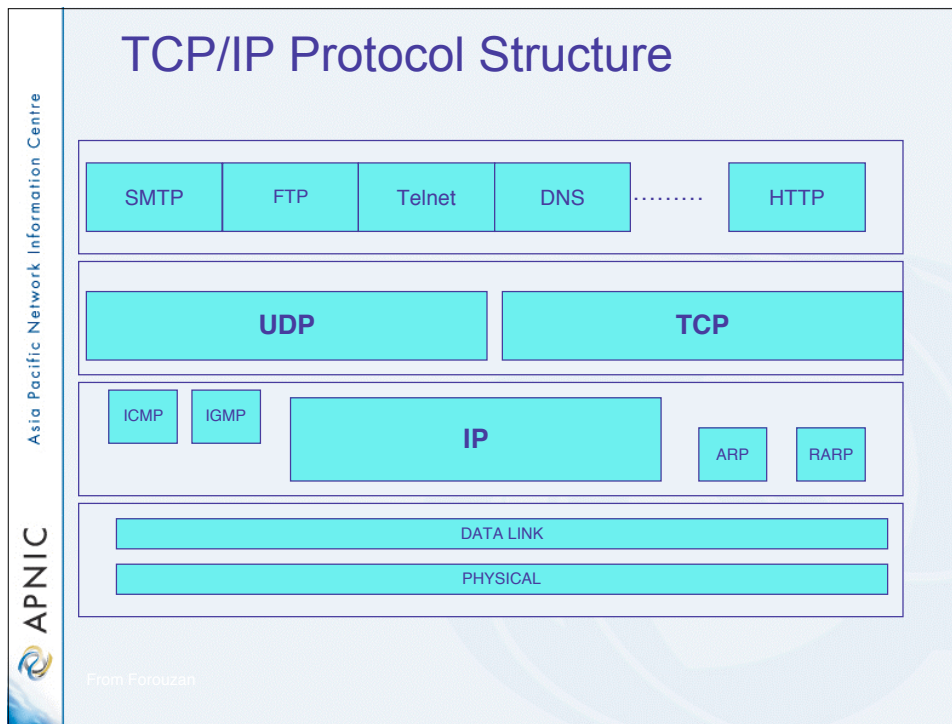
- The Protocols implement the common communication processes used to send all data for all services (applications)
  - In an internetwork then, all data, for whatever application being used, is seen and sent in the same way.
- It is the protocols on each end that determine which application is the one the data relates to and then makes this data available to that application



APNIC  
Asia Pacific Network Information Centre

To the heart of it.....

IP



- Asia Pacific Network Information Centre
- ## Internet Protocol (IP)
- IP is an unreliable, connectionless delivery protocol
    - A best-effort delivery service
    - No error checking or tracking (no guarantees – Post Office)
    - Every packet treated independently
      - Can follow different routes to same destination
    - IP leaves higher level protocols to provide reliability services (if needed)
  - IP provides three important definitions:
    - basic unit of data transfer
      - specifying exact format of the headers
    - routing function
      - choosing path over which data will be sent
    - rules about delivery
      - how IP datagrams should be processed
      - how to deal with unusual events (errors)
- APNIC



## IP Datagram format

- That part of a packet containing the IP headers and the data from the higher layers passed to the IP layer are called **datagrams**
- IP specifies the header information for the data it requires for its tasks - information needed for routing and delivery
  - eg source and destination IP addresses
- It has nothing to do with higher layer headers or data and can transport arbitrary data
- Header is 20 to 60 bytes

**Datagram header**

**Datagram data area**

## IP Datagram header fields

VER 4 bits	HLEN 4 bits	DS (TOS) 8 bits	Total length 16 bits	
Identification 16 bits			Flag S 3 bits	Fragmentation offset 13 bits
Time to live 8 bits	Protocol 8 bits		Header checksum 16 bits	
Source IP address				
Destination IP address				
Options				

## IP Datagram fields

- **VER**

- 4-bit field
- Version of IP Protocol used to create datagram
  - currently IPv4
- used to verify that sender, receiver, and any routers in between agree on datagram format

- **HLEN**

- 4-bit field
- Gives datagram header length (variable)
- measured in 32-bit (4-byte) words
- all header fields have a fixed length except IP options field
- most common header length (minimum) is 20 bytes
- Maximum is 60 bytes

## Datagram fields

- **Differentiated Services (RFC 2474)**

- IETF introduced a change in this field:
  - change name of ToS field to *Differentiated Services*
  - provide a different interpretation for the 8-bits
- First 6 bits make up a **codepoint**
  - When 3 right-most bits are 0s:
    - 3 left-most bits are precedence bits for compatibility with old Service Type interpretation
  - When 3 right-most bits are not 0s:
    - 6 bit value represents 64 service types (not yet finalised) divided into three categories representing assigning authority:
      1. Internet (32 service types)
      2. Local (16 service types)
      3. temporary/experimental (for testing – 16 service types)

Last 2 bits not used



APNIC  
Asia Pacific Network Information Centre

## Datagram fields

- **TOTAL LENGTH field**
  - 16 bit field
  - Defines total length (header plus data) of IP datagram measured in bytes
  - maximum size of IP datagram is 65535 octets
  - Length of data = total length – header length
    - Eg. HLEN = 5 (4-byte words) = 20 bytes
    - Total length = 24 (bytes)
    - Length of data = 4 bytes

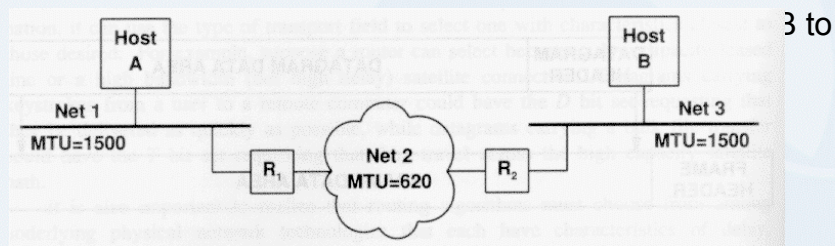
APNIC  
Asia Pacific Network Information Centre

## Datagram fields

- **3 Fragmentation fields**
  - IDENTIFICATION
    - 16-bit field
  - FLAGS
    - 3-bit field
  - FRAGMENT OFFSET
    - 13-bit field
- All used to control fragmentation of datagrams
- A datagram has a max size of 65,535
- Must be carried inside a physical network frame
- If this datagram then cannot be accommodated it is then *fragmented* or split up

## Fragmentation example

- Fragmentation is carried out by routers (in IPv4-different in IPv6)
  - Router R1 fragments large datagrams sent from A to B



## Fragmentation

- Each fragment has its own header – It is a separate datagram
  - Most fields repeated and some changed
- A fragment can be further fragmented if it encounters even smaller MTUs
- IP datagram fragments are only reassembled at the final destination
- Because fragments are carried as independent packets they may arrive in any order (or not at all)




## Datagram fields

- **TIME TO LIVE**
  - 8-bit field
  - Specifies how long (in seconds) datagram is allowed to remain on the Internet
  - Protects the Internet from infinite looping
  - Each router that processes a datagram decrements TTL count by:
    - 1 (representing a HOP)
    - or the time a packet spends at router if there is a long delay at a router (usually not the case now)
- If TTL reaches 0 router discards the datagram

## Datagram fields


- **PROTOCOL field**
  - 8 bit field
  - similar to type field in a network frame
  - value specifies which high-level protocol was used to create message carried in data area of datagram
  - Examples:
    - ICMP 1
    - TCP 6
    - UDP 17



Asia Pacific Network Information Centre

## Datagram fields

- **HEADER CHECKSUM field**
  - 16 bit field
  - Ensures the integrity of header values
  - Treats header as a sequence of 16-bit sections
    - adds sections together with 1's complement and takes the 1's complement of the result
  - Checksum is calculated by sender and value obtained is sent with packet
  - Receiver repeats same calculation to check packet is unaltered



Asia Pacific Network Information Centre

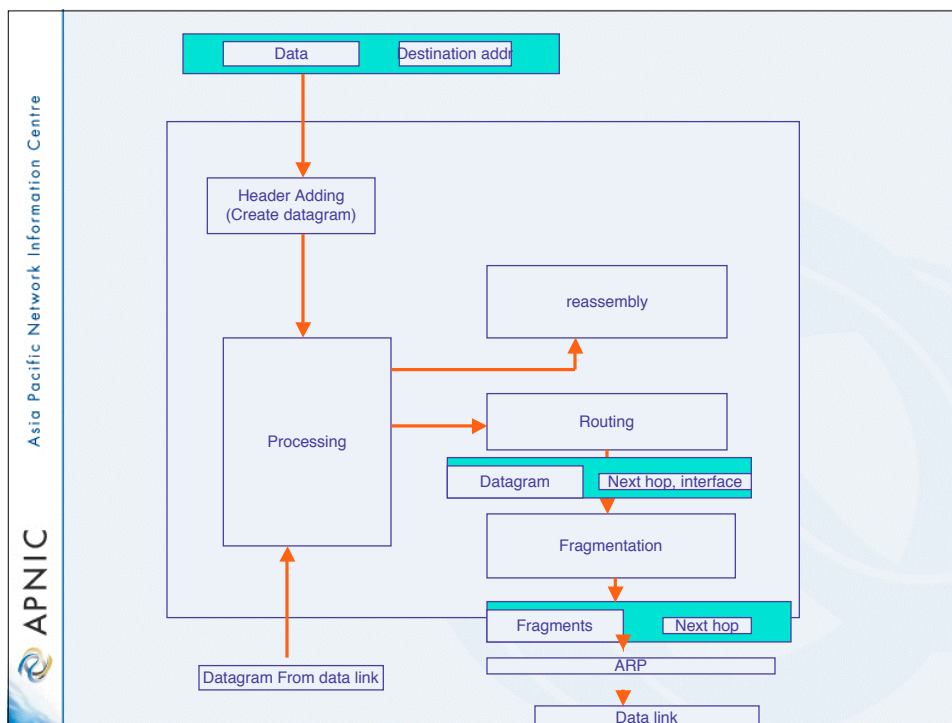
## Datagram fields

- **OPTIONS field (Optional field)**
  - Allows for optional services for the datagram
  - Used for network testing and debugging
  - There are currently:
    - two single byte options
  - Four multiple byte options
- **Maximum 40 bytes**



## The IP Process

- Header adding
- Processing
- Routing
- Fragmentation
- ARP and delivery
- Reassembly

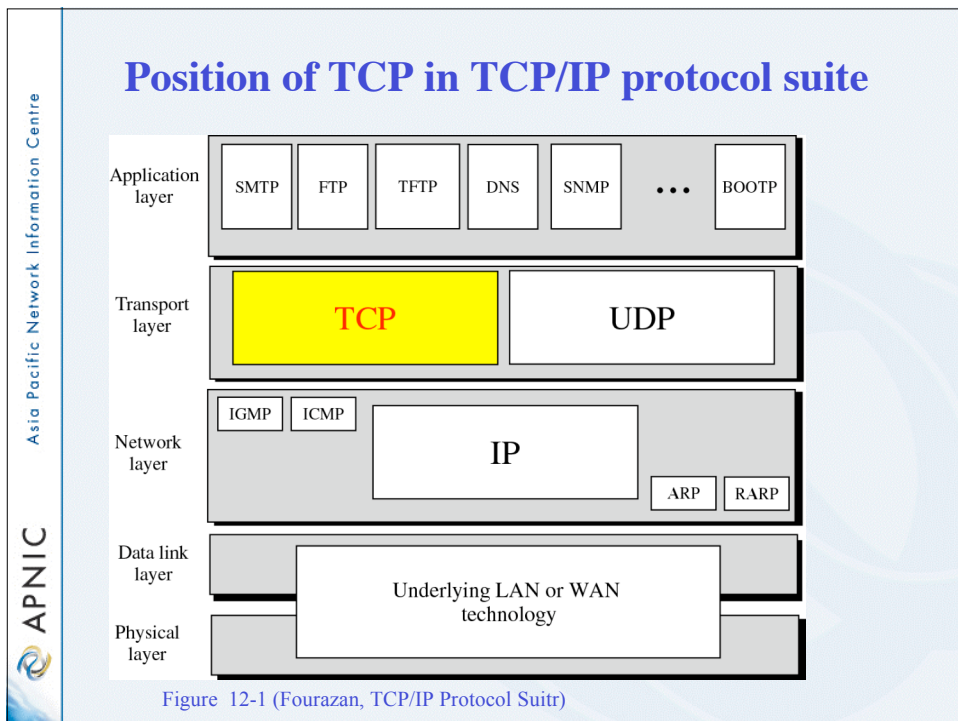


APNIC  
Asia Pacific Network Information Centre

# TCP

## (Transmission Control Protocol)

RFC 793





### Reliable Stream Transport Service

#### TCP

- TCP provides a **reliable transport service** for IP
- It removes the need for application programmers to build complex transport (reliability and sequencing) software.
  - Functional commonality
- It is **connection based** and **reliable**
- Provides a virtual connection in a connectionless environment

### Providing Reliability

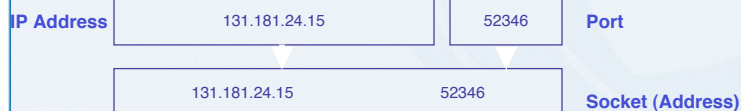
- A reliable stream delivery guarantees delivery of a stream of data from one machine to another **without duplication or loss**
- To achieve this TCP uses  
**positive acknowledgement with retransmission**
- The technique requires the destination to send back an acknowledgement (ACK) to the source when data has been successfully received

## Ports, Connections and End Points

- TCP uses **port numbers** to identify the ultimate destination (the application) within a machine
- However, because TCP establishes a **virtual connection**, the port number alone is not enough to ensure uniqueness for every connection
  - Multiple machines (with the same ephemeral port) could send to the same port on a server

## Sockets

- TCP then uses the **IP address** as well as the port number to establish a unique identifier on a machine - this is defined as a socket or end point or socket address.
  - Consider it is a socket into which a connection is “plugged”
- Every connection has 2 sockets or endpoints:
  - Source IP and Source Port
  - Destination IP and Destination Port
- The combination of the 2 sockets makes every connection unique.





## Passive and Active Opens

- TCP requires both endpoints to agree to participate - they must first establish the connection
  - This is to synchronise start parameters
- The application program at one end performs a **passive open** function by calling TCP and indicating that it will accept incoming connections (a TCP port number is pre-defined)
  - (server) – It waits for a connection
- The application program at the other end calls its TCP using an **active open** to establish a connection to a the server. (A ephemeral TCP port number is assigned)- (client). It initialises the connection

## Passive and Active Opens

- The connection is *full duplex*
  - Information (data) can be carried in both directions
- The two TCP software modules communicate to establish and verify the connection
- Once a connection has been created, application programs can begin to pass data directly through the connection

## Transmission Control Protocol

- What does TCP define?
  - the format of the data (headers) and information that two computers exchange to achieve a reliable transfer
  - procedures used to ensure the data arrives correctly and is passed to the correct application
  - It specifies how TCP software distinguishes among multiple destinations on a given machine
  - It specifies how two computers initiate a TCP stream and how they agree when it is complete

## TCP Segment Format

- The unit of transfer between the TCP software on two machines is called a **segment**

0	4	10	16	24	31
SOURCE PORT			DESTINATION PORT		
SEQUENCE NUMBER					
ACKNOWLEDGEMENT NUMBER					
HLEN	RESERVED	CODE BITS	WINDOW		
CHECKSUM			URGENT POINTER		
OPTIONS (IF ANY)				PADDING	
DATA					
...					



## Establishing a TCP Connection

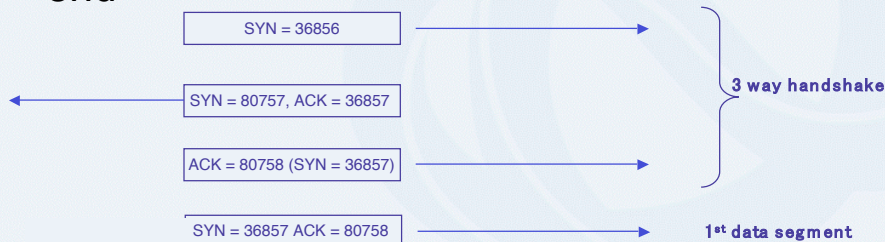
- Because TCP is *connection orientated* it needs to first initialise a virtual connection
- This connection remains open for the duration of the interaction between the 2 ends and is used to stream the data
- TCP uses a **three-way handshake to initialise and synchronise the connection**
  - Essentially this sets up the initial sequence numbers at both ends and creates a **Transmission Control Block (TCB)** for that connection (later)

## Establishing a TCP Connection

- The first message of the handshake has the **SYN** bit set in the code field
- The second message has both the **SYN** bit and **ACK** bit set, indicating it is acknowledging the first **SYN** message – It in fact combines two messages
  - TCP can “piggy back” ACK messages
- The final message has only an **ACK** bit

## Initialisation

- Note that the initial sequence number is a random value.
- It represents the start value for the data being sent.
- Assuming a syn is initialised to 36856 by the first end and 80757 by the second end



## Closing a TCP Connection

- TCP uses a **modified three-way handshake** to close connections.
- An application (client) tells TCP it has no more data, TCP will close the connection **for that direction** (it will send out a **FIN** message)
- The receiver will acknowledge this message and then notify its application.
- That application will then close and the receiving TCP will send its own FIN message
- This will be separately acknowledged by the sender



## Timers

- TCP uses a retransmission timer (there are other timers as well)
- Retransmission Timer
  - Every time the sender sends a segment it sets a timer for that segment
  - This timer is calculated to be  $2 \times \text{RTT}$  (round trip time)
  - If an acknowledgment for a segment is not received by the time the timer has expired the sending TCP retransmits the segment

## Positive acknowledgement and flow control

- Every segment received carries a chunk of data from the data stream with a sequence number indicating the position in the data stream of the first byte in that segment.
- TCP acknowledges every byte of data sent over a TCP connection by sending an acknowledgment number which is the next byte the receiver is expecting to receive
- The Acknowledgement number confirms that all bytes to that number have been successfully received
- Individual segments are not necessarily acknowledged.

## Sequence Numbers -Example

A TCP connection is transferring a file of 6000 bytes. The first byte is numbered 10010. What are the sequence numbers for each segment if data is sent in five segments with the first four segments carrying 1,000 bytes and the last segment carrying 2,000 bytes?



## Sequence Numbers

The following shows the sequence number for each segment:

Segment 1	➔	10,010	(10,010 to 11,009) (1000)
Segment 2	➔	11,010	(11,010 to 12,009) (1000)
Segment 3	➔	12,010	(12,010 to 13,009) (1000)
Segment 4	➔	13,010	(13,010 to 14,009) (1000)
Segment 5	➔	14,010	(14,010 to 16,009) (2000)

Sequence numbers are used together with acknowledgment numbers to manage the data stream and to ensure that all Data sent is received and passed to the application in the correct order



## Flow Control - Sliding Windows

- A simple positive acknowledgement protocol wastes a substantial amount of network bandwidth because it must delay sending a new packet until it receives an acknowledgement for the previous packet

## Sliding Windows

- **TCP implements a Sliding windows** protocol that allows the sender to transmit multiple bytes before waiting for an acknowledgement.
  - The receiver advises the amount of data it is able to receive (current space available in buffer).
    - This is set as the window and data in the window can be sent without waiting for an acknowledgement
  - Bytes not acknowledged are kept in the window (sender buffer) until acknowledged so these can be retransmitted if there is a problem
    - The window “slides” as data is acknowledged and window advertisements received – passing over the acknowledged data to include new data that can now be sent.
  - Processing speeds are accommodated
  - Network speed is accommodated


## Congestion Control

- Congestion can occur on a network when a router cannot process packets received and discards these - causing retransmissions and potentially further congestion
- TCP assumes lost segments to be caused by network congestion (discards)
- A network therefore, that cannot deliver data as fast as it is being sent, should be able to also tell the sender to slow down - ie to reduce the sender window.

## Congestion Window size


- TCP uses a congestion window size as well as the receiver-advertised window size to determine the actual sender window.
  - $\text{actual window} = \min(\text{rec\_wind}, \text{congest\_wind})$
- TCP uses a strategy of *slow start and additive increase* and *multiplicative decrease* to adjust the congestion window size



APNICAsia Pacific Network Information Centre


## Error Control

- This is based on
  - The checksum
  - Acknowledgment
  - Time-out

APNICAsia Pacific Network Information Centre

## Corrupted Segment


- Receiver checks checksum finds segment to be corrupt
- Discards segment and does not send an ack
- Sender waits and when timer for that segment expires without an ack having been received, resends the segment



APNIC  
Asia Pacific Network Information Centre

## Lost Segment

- No segment is received so no ack is sent
- Sender waits and when timer for that segment expires without an ack having been received, resends the segment



APNIC  
Asia Pacific Network Information Centre

## Duplicate Segment

- A segment can be resent if there is a delay in receiving the ack
- The receiver will simply discard a segment if it has already received a segment with the same sequence number



## Out of Order Segment

- The receiver will only acknowledge a segment if it has received all the segments that precede it. That is: in order.
- If the ack is therefore delayed the sender may send a duplicate which is handled as previously discussed

## Lost ACK

- This might cause a duplicate to be sent or might not even be noticed.
- Acks are accumulative:
  - If ack 1601 is lost but ack 1801 is received, all the data to 1800 is assumed to have been received

## Out of Band Data

- Although TCP is a stream-oriented protocol, it is sometimes important for the program at one end of a connection to send data **out of band (out of sequence)**
- TCP allows the sender to specify data as **urgent**
  - It sets the URG flag
  - Inserts the urgent data at the beginning of a segment
  - Sets the URGENT pointer to point to the first byte of “normal” data inserted after the urgent data
- When the receiver receives the segment with the URG flag set, it extracts the urgent data (using the urgent pointer ) and immediately passes it to the receiving application

## PSH Flag

- To ensure that data is sent and processed immediately, (interactive data), the sender client can request a PUSH operation
- The sending TCP immediately sends this data (does not wait for Window to be filled -).
- It also sets the PSH flag
- When the receiving TCP receives this packet with the PSH flag set, it immediately passes it to the receiving application
- The PSH flag usually is implemented by default

## RST Flag

- When a segment with the RST flag set is received, the TCP connection is terminated.
- Is used to if
  - A connection to an unknown port is requested. The RST cancels the connection
  - One end may need to end the connection (it may detect an unusual state)
  - A TCP connection has been idle for too long



## Transmission Control Blocks

- TCP **maintains** its connections by keeping a table that holds information about each connection
- This is known as the TCB Table and is updated after every transmission/receipt of a segment to reflect the current values and **state** of the connection
- A separate TCB is established for every connection

## The TCB contains.....

- State (one of eleven finite states)
- The Process using this connection
- Local IP
- Local Port
- Destination IP
- Destination Port
- Interface
- Local window
- Remote Window
- Sending sequence number
- Receiving sequence number
- Sending ACK number
- RTT
- Time out values – when to retransmit
- Buffer size
- Buffer pointer

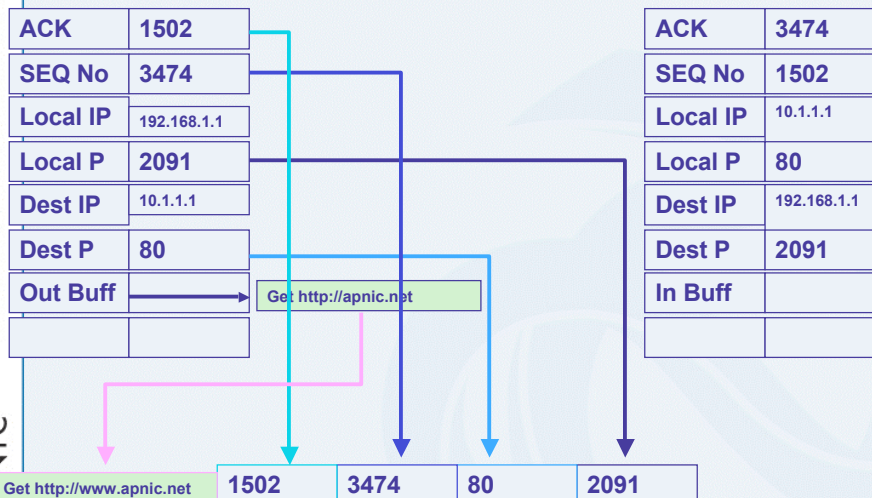
## TCP Processing

- When a NW application starts it will create a TCB block (Initially Closed) defining its end of the connection and containing the initial set up information
- When the 3-way handshake is initialised, the receiving end also creates a corresponding TCB
- When the connection is made, the state of the connection is defined as established and the entry updated in the TCBs on both ends.

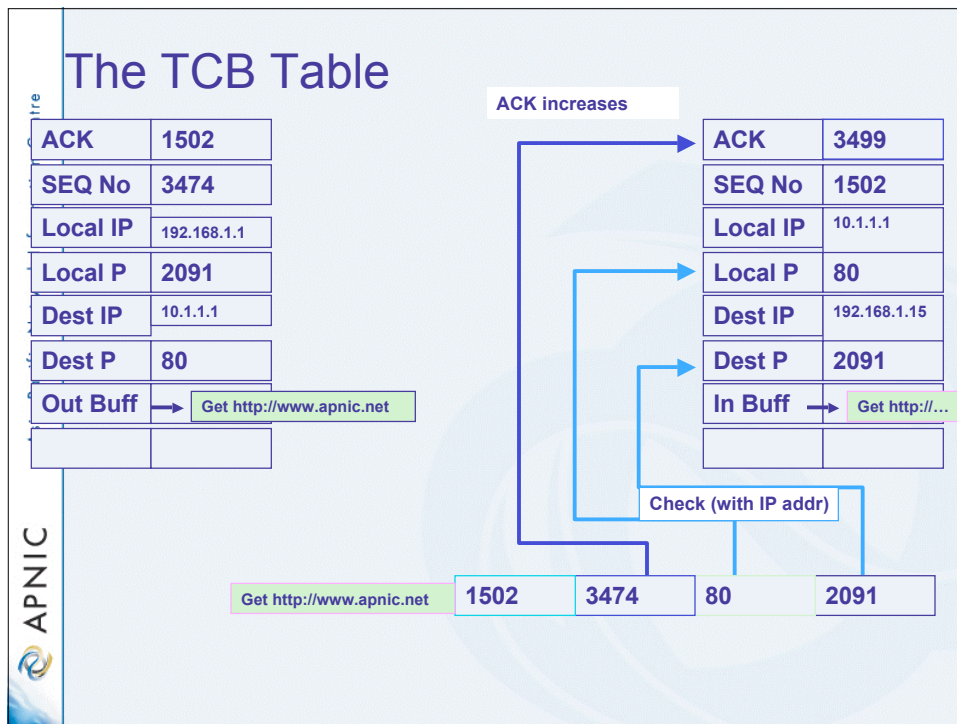
Data can now be “streamed” between the 2 ends

- When a TCP segment is received the TCB is searched for a corresponding TCB with the the socket addresses in the segment, and the information in the TCB used to process the incoming data
- After this processing the TCB is updated to reflect the current state of the connection.

## The TCB Table -Example







- ## TCB States
- There are 11 states
    - CLOSED – no connection
    - LISTEN – server is waiting
    - SYN-SENT – connection request sent, waiting for ack
    - SYN-RECVD – connection request received
    - ESTABLISHED – connection established
    - FIN-WAIT-1 – the app has requested to close the connection
    - FIN-WAIT-2 – The other side has accepted the close
    - CLOSING – both sides close simultaneously
    - TIMED-WAIT – after closing a connection TCP waits before deleting the connection so that any duplicate segments are removed and cannot interfere in a new connection. Continues to acknowledge.
    - CLOSE-WAIT – server is waiting for app to close
    - LAST-ACK – server is waiting for last acknowledgment

## Analysing Data –Using a Protocol Analyser

- **Analysing Data**
- Data making up a frame is sent as a series of bits. It is the protocols that allow various processes to differentiate between different fields and components in this bit stream.
  - For example; the IP protocol defines the field for the destination IP address so that the process can extract this information and use it to forward the packet. It also identifies where the data it is carrying (the TCP segment in this case) begins.
- When packets (or frames) are captured using packet capture software such as Wireshark, the bit stream making up that frame is displayed as a string of hexadecimal digits (showing this in binary would make it too unmanageable)

## Protocol Analysers - Wireshark

- A packet is analysed based on the protocols defined and packet contents (header fields and data fields making up the packets layers) are displayed in a hierarchical fashion from the bottom up.
- Wireshark allows you to view all layers making up a packet and to expand each layer to view individual fields and their contained values. This is how data transfers can be analysed.
- Note: While in reality devices also generate bits to control transmission, (preamble, end bits, check bits) the packet capture program will generally not display this physical network dependent information and will start with the hardware destination address in the frame.



## Wireshark

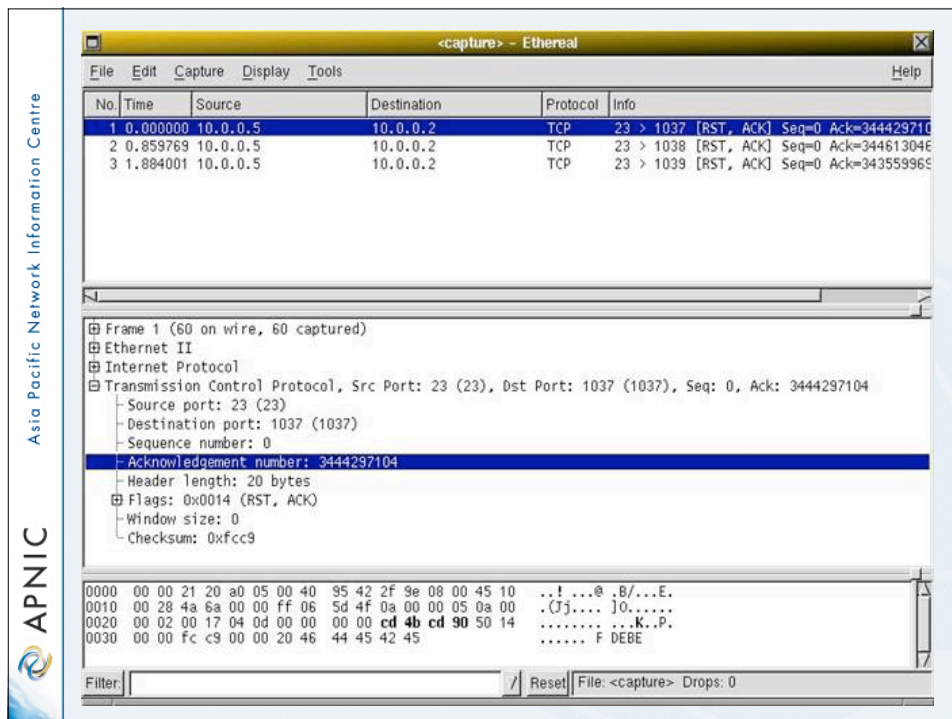
- **Wireshark (Ethereal)**
- Wireshark is a free network protocol analyser for Unix and Windows.
- It allows you to examine data from a live network or from a capture file on disk.
- Frames read off the wire are stored and analysed by applying protocol decoders or dissectors –
  - These use protocol “rules” to decode the bits in the packet and displays this data
- It can be downloaded from <http://www.wireshark.org/>.
- Documentation and complete user guide can be downloaded from:  
<http://www.wireshark.org/docs>

## Using Wireshark

- **Wireshark allows you to capture packets as they are transmitted on the network medium and view these in real time or from a capture file.**
- Wireshark's display window is comprised of three main windows, or panes:
  - The top pane is the **packet listing pane**. It displays a summary of each packet captured. By clicking on packets in this pane you control what is displayed in the other two panes.
  - The middle pane is the **tree view pane**. It displays the packet selected in the top pane in more detail by layer and allows you to expand each layer to view the content field values
  - The bottom pane is the **data view pane**. It displays the data in HEX from the packet selected in the top pane, and highlights the field selected in the tree view pane.

## Wireshark Layout

- By clicking on a packet in the Packet Listing panel selects that packet in the other 2 panes.
- You can then expand any part of the packet, layer by layer, by clicking on the “+” sign in the tree view.
- This will show the contents of the header fields and their meanings.
- The selected header is also shown highlighted in the dataview

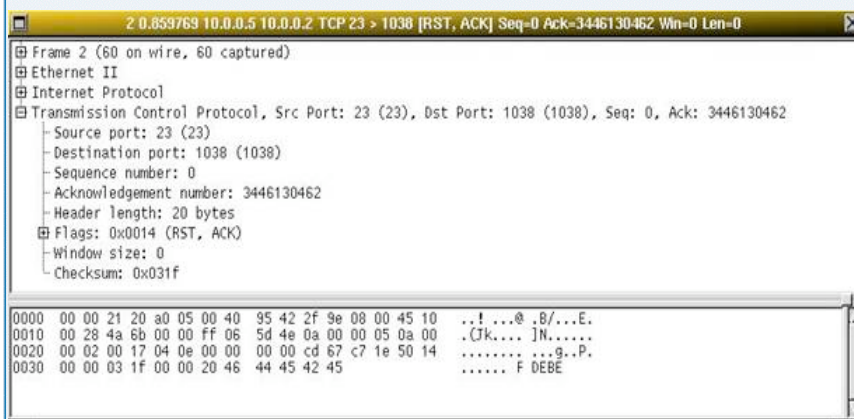





## View packet in a separate window

By selecting a packet in the packet list pane, and then choosing the *"Show Packet in New Windows"* option in the Display menu, you can view packets in separate window.

## Packet Window






APNIC  
Asia Pacific Network Information Centre

## Wireshark Features

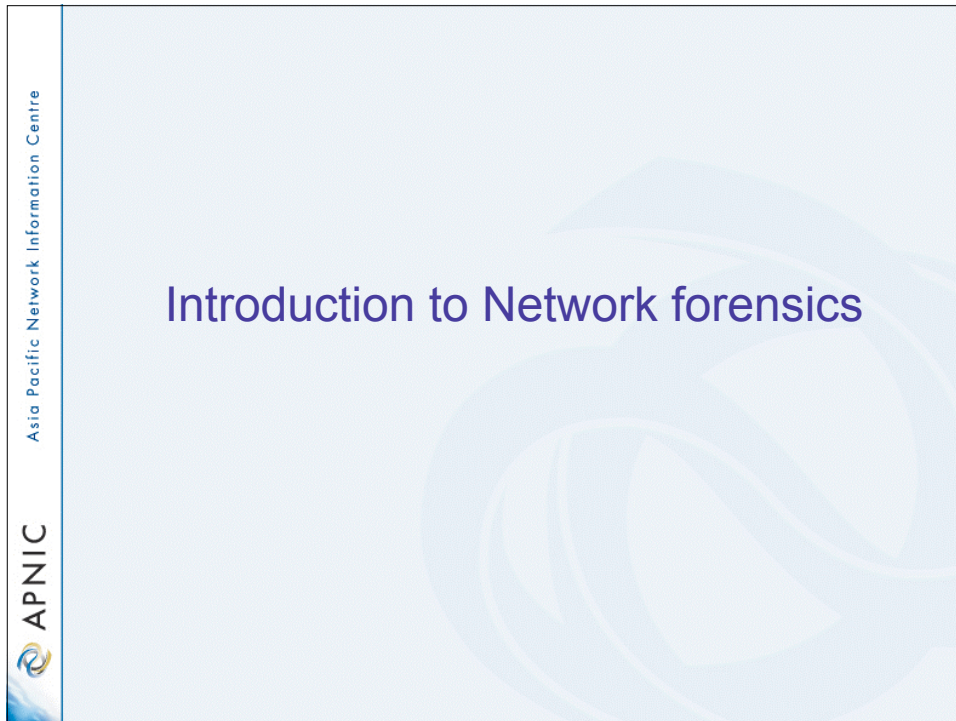
- Capture packets in real time
- Save captures and review using single or multiple files
- Apply capture filters or display filters
- Colourise packets
- Search for packets
- Review statistics



APNIC  
Asia Pacific Network Information Centre

## Wireshark Demo



This slide has the same design as the first one, with a light blue background and a vertical blue bar on the left containing the APNIC logo and text. The title 'Digital forensics overview' is centered in a dark blue font. Below the title is a bulleted list:

- Forensics science
  - The application of science to the law
- Digital forensics (forensics)
  - Also know as
    - Computer and network forensics
  - Application of science to the
    - Identification of data
    - Collection of data
    - Examination of data
    - Analysis of data

At the bottom right of the slide, there is a small reference text: 'Ref: Guide to Integrating Forensic Techniques into Incident Response'.

## Digital forensics overview

- Digital forensics techniques can be used for many purposes
  - Investigating crimes – evidence collection for legal proceedings
  - Internal policy violations – internal disciplinary actions
  - Reconstructing computer security
  - Troubleshooting operational problems – including handling of malware incidents
  - Recovering from accidental system damage
- Without such capability, it will be difficult to determine:
  - What has happened?
  - What damage is incurred
  - Who caused the problem?
  - How did it happen?
  - How can the problem be rectified
  - How to prevent future incidents

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Process of digital forensics

- Collection
  - Identifying, labelling, recording, and acquiring data from all possible sources while preserving data integrity
- Examination
  - Processing collected data forensically, and assessing and extracting data of particular interest while preserving data integrity
- Analysis
  - Analysing the results of examination, using legally justifiable methods and techniques
- Reporting
  - Reporting the results of the analysis, providing recommendations for improvement of policies, procedures, tools, and other aspects of forensic process

Ref: Guide to Integrating Forensic Techniques into Incident Response



## Collection of data

- Identifying possible data sources
  - Typically, desktop computers, servers, network storage devices, and laptops, PDAs, cell phones, digital cameras, digital recorders, audio players and etc.
  - Possible data sources located in other places
    - E.g., Network activity and application usage within an organisation
  - Information may be recorded by other organisations
    - E.g., ISPs

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Collection of data

- Analysts should be:
  - mindful of the owner of each data source and the effect on collecting data
    - E.g., getting copies of ISP records typically requires a court order
  - aware of the organisation's policies and legal considerations regarding externally owned property at the organisation's facilities and locations outside the organisation's control
    - E.g., employee's personal laptop, a contractor's laptop
    - E.g., a computer at a telecommuter's home office

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Collection of data

- Some useful methods to collect data
  - Keep audit records
    - E.g., Most OSs can be configured to audit and record certain types of events
  - Centralised logging
    - Certain systems and applications forward copies of their logs to secure central log servers
    - Security monitoring controls (E.g., intrusion detection software, anti-virus software, and spyware detection and removal utilities) can generate logs of attacks and intrusions
  - Monitoring of user behaviour
    - Keystroke monitoring
      - Be aware this is a violation of privacy unless users are advised through organisational policy and login banners
      - Employing such method should be discussed with legal advisors and documented clearly in the organisation's policy

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Acquiring the data

- Analyst should make a informed decision regarding the prioritisation of data source acquisition
  - Develop a plan to acquire the data
    - Consider likely value, volatility of data and amount of effort required
  - Acquire data
    - Can be acquired through security tools, analysis tools, or other means
    - Can be acquired through forensic tools
  - Verify the integrity of the data
    - Important to prove that the data has not been tampered
    - Can use tools such as message digest

Ref: Guide to Integrating Forensic Techniques into Incident Response



## Acquiring the data

- A clear defined chain of custody should be followed to avoid allegations of mishandling or tampering of evidence
  - Keeping a log of every person who had physical custody of the evidence, documenting the actions performed on the evidence and time
  - Storing the evidence in a secure location
  - Making a copy of the evidence and performing examination and analysis using only the copied evidence
  - Verifying the integrity of the original and copied evidence

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Data sources

- Using data from data files
- Using data from Operating Systems
- Using data from network traffic
- Using data from applications
- Using data from multiple sources

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Data sources

- Network forensics analysis relies on all of the layers
- Hardware layer (=Data link layer) provides information about physical components
- Other layers describe logical aspects
- An analyst can map an IP address (logical identifier at the IP layer) to the MAC address (Media Access Control) of a particular NIC (Network Interface Card = physical identifier at the physical layer)
  - An analyst can identify a host of interest
  - Identifying a host helps to identify most likely being used applications

Ref: Guide to Integrating Forensic Techniques into Incident Response

## Network Forensics



### Team Cymru

So, now, we need to make the bad guy's life more difficult.

Objective: deter miscreants from committing online crime.

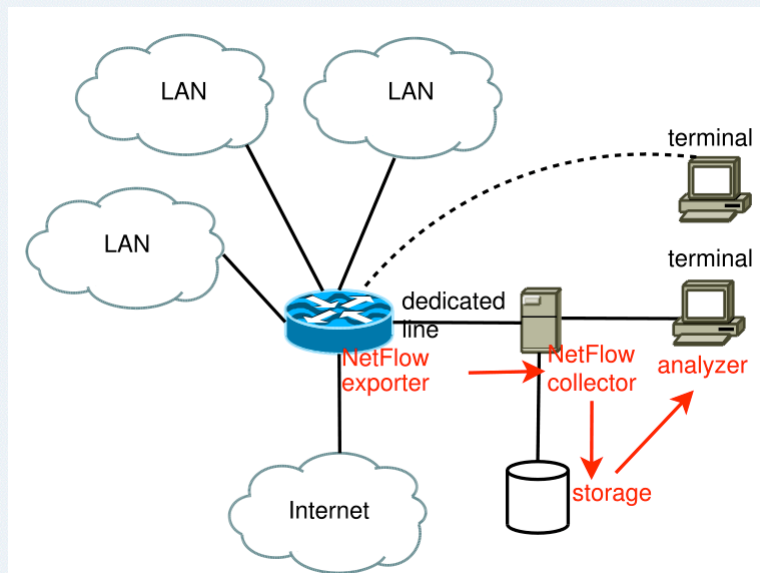


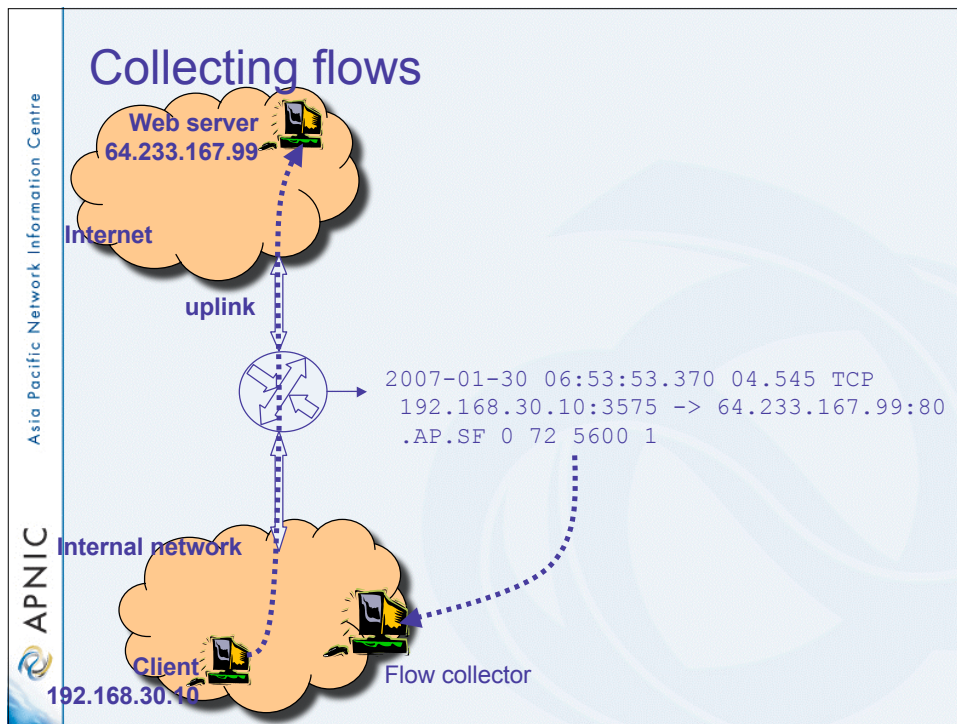
## Botnets - How do we find them?

*Network Forensics*

- (1) Watch flows
- (2) Watch DNS
- (3) Effectively use Darknets
- (4) Sniffing
- (5) Sandboxing
- (6) Malware analysis

## Flow collection





Asia Pacific Network Information Centre

## Collecting flows – enabling collection

A generic Cisco example:

```
interface fastethernet 0/0
ip route-cache flow
```

Set to netflow version 5 and set timeout:

```
ip flow-export <ip> <port>
ip flow-export version 5
```

Break-up long flows into 5 minute segments (should be less than your file rotation time):

```
ip flow-cache timeout active 5
```

APNIC



## Collecting flows – enabling collection

### nfcapd

- Flow collector
- Listens for flows on a given port and stores the data into files that are rotated a pre-set number of minutes
- One nfcapd per flow stream
- Example:

```
nfcapd -w -D -l /var/log/flows/router1 -p 23456
```

```
nfcapd -w -D -l /var/log/flows/router2 -p 23457
```

-w: sync file rotation with next 5 minute interval

-D: fork to background

-l: location of log file

## Collecting flows – enabling collection

- May wish to use nfdump on the resulting files to insert flow records into a database
- *Stager*: system for aggregating and presenting network statistics.
  - Collects & stores network info (netflow, SNMP, MPing) in a database
  - Provides a web front-end

## Watching flows

*Total network awareness*

The diagram illustrates the mapping of network flow data fields to a specific data row. The data row is: 2025-08-30 06:53:53.370 63.545 TCP 113.138.32.152:25 -> 222.33.70.124:3575 .AP.SF 0 62 3512 1. Labels with arrows point to each field: Date, Duration, Source IP:Port, TCP flags, Packets, Flows, Start time, Protocol, Destination IP:Port, Type of Service, and Bytes.

Field	Value
Date	2025-08-30
Start time	06:53:53.370
Duration	63.545
Protocol	TCP
Source IP:Port	113.138.32.152:25
Destination IP:Port	222.33.70.124:3575
TCP flags	.AP.SF
Type of Service	0
Packets	62
Bytes	3512
Flows	1

## Watching flows

Sort flows by total number of bytes

Packets	Bytes	pps	bps	Bpp	Flows
1.4 M	2.0 G	2023	5.6 M	1498	1

```
nfdump -r nfcapd.200508300700
-o extended -s srcip -s ip/flows
-s dstport/pps/packets/bytes
-s record/bytes
```

Time	Flow	Prot	Src IP Addr:Port	Dst IP Addr:Port	Flags	Tos	Packets	Bytes	pps	bps	Bpp	Flows
2005-08-30	TCP	126.52.54.27:47303	->	42.90.25.218:435	.....	0	1.4 M	2.0 G	2023	5.6 M	1498	1
2005-08-30	TCP	198.100.18.123:54945	->	126.52.57.13:119	.....	0	5673732	795.1 M	627	2.5 M	1468	1
2005-08-30	TCP	126.52.57.13:45633	->	91.127.227.206:119	.....	0	321148	456.5 M	355	4.0 M	1490	1
2005-08-30	TCP	126.52.57.13:45598	->	91.127.227.206:119	.....	0	320710	455.9 M	354	4.0 M	1490	1
2005-08-30	TCP	126.52.57.13:45629	->	91.127.227.206:119	.....	0	317764	451.5 M	351	4.0 M	1489	1
2005-08-30	TCP	126.52.57.13:45634	->	91.127.227.206:119	.....	0	317611	451.2 M	351	4.0 M	1489	1
2005-08-30	TCP	126.52.57.13:45675	->	91.127.227.206:119	.....	0	317319	451.0 M	350	4.0 M	1490	1
2005-08-30	TCP	126.52.57.13:45619	->	91.127.227.206:119	.....	0	314199	446.5 M	347	3.9 M	1490	1
2005-08-30	TCP	126.52.54.35:59898	->	132.94.115.59:2466	.....	0	254717	362.4 M	322	3.7 M	1491	1
2005-08-30	TCP	126.52.54.35:59773	->	55.107.224.187:11709	.....	0	272710	348.5 M	301	3.1 M	1340	1

...the possibilities are endless...



Asia Pacific Network Information Centre  
 APNIC

## Watching flows

### nfdump

```

nfdump -r nfcapd_file
-A src,dstport
-c 10 'src ip 192.168.2.12'
    
```

See scanning on your network...

Date	Time	flow	start	Prot	Src IP Addr:Port	Dst IP Addr:Port	Packets	Bytes
2006-12-02	14:02:12	TCP			192.168.2.12:47303 -> 192.168.2.13:445		1	60 B
2006-12-02	14:02:12	TCP			192.168.2.12:47304 -> 192.168.2.14:445		1	60 B
2006-12-02	14:02:12	TCP			192.168.2.12:47305 -> 192.168.2.15:445		1	60 B
2006-12-02	14:02:12	TCP			192.168.2.12:47306 -> 192.168.2.16:445		1	60 B
2006-12-02	14:02:12	TCP			192.168.2.12:47307 -> 192.168.2.17:445		1	60 B
2006-12-02	14:02:13	TCP			192.168.2.12:47308 -> 192.168.2.18:445		1	60 B
2006-12-02	14:02:13	TCP			192.168.2.12:47309 -> 192.168.2.19:445		1	60 B
2006-12-02	14:02:13	TCP			192.168.2.12:47310 -> 192.168.2.20:445		1	60 B
2006-12-02	14:02:13	TCP			192.168.2.12:47311 -> 192.168.2.21:445		1	60 B
2006-12-02	14:02:13	TCP			192.168.2.12:47312 -> 192.168.2.22:445		1	60 B

Asia Pacific Network Information Centre  
 APNIC

## Watching flows

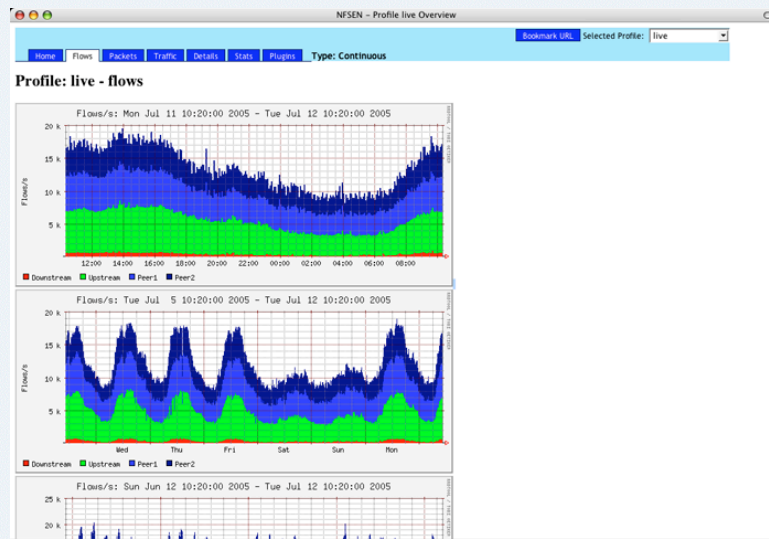
### nfsen – a graphical interface!

The screenshot shows the nfsen web interface with a navigation bar (Home, Flows, Packets, Traffic, Details, Stats, Plugins) and a 'PortTracker' section. It features several small graphs for TCP Packets, TCP Bytes, UDP Flows, UDP Packets, and UDP Bytes. The main graph is titled 'Med Jul 13 16:10:00 2005 - Thu Jul 14 16:10:00 2005 - TCP Flows' and displays a large volume of traffic over time. On the right, there are controls for 'Show Top 10 Ports', 'Graph' (Linear, Log, Stacked), and 'Track Ports' (Add, Delete).

<http://nfsen.sourceforge.net>

## Watching flows

### nfsen – a graphical interface!



<http://nfsen.sourceforge.net>

## Watching flows

### Identify DDoS sources

DDoS sources are very likely compromised devices (assuming they aren't spoofed).

