EC-Council

C|CT

Certified | Cybersecurity Technician

Module - 17

Network Traffic Monitoring

This page is intentionally left blank.

Module Objectives

Organizations need to perform network monitoring and analyze network traffic to identify suspicious activities across their networks. This module covers the concept of network traffic monitoring.

At the end of this module, you will be able to do the following:

- Understand the need for and advantages of network traffic monitoring
- Understand the network traffic signatures
- Describe the categories of suspicious traffic signatures
- Explain the attack signature analysis techniques
- Understand network monitoring for suspicious traffic
- Understand the various network monitoring tools

Understand the Need and Advantages of Network Traffic Monitoring

The objective of this section is to explain in detail the need for and advantages of network traffic monitoring.

Network Traffic Monitoring

Network traffic monitoring is the process of capturing network traffic and inspecting it closely to determine what is happening on the network. Network monitoring is a retrospective security approach that involves monitoring a network for abnormal activities, performance issues, bandwidth issues, etc. It is an integral part of network security and is a demanding task within the network security operations of organizations. Continuous network traffic monitoring and analysis are required for effective threat detection. Security Professional should constantly strive to maintain smooth network operation. If a network goes down even for a small period, productivity within a company may decline. To be proactive rather than reactive, the traffic movement and performance must be monitored to ensure that no security breach occurs within the network.

The network monitoring process involves sniffing the traffic flowing through the network. For this purpose, network packets must be captured, and a signature analysis must be conducted to identify any malicious activity.

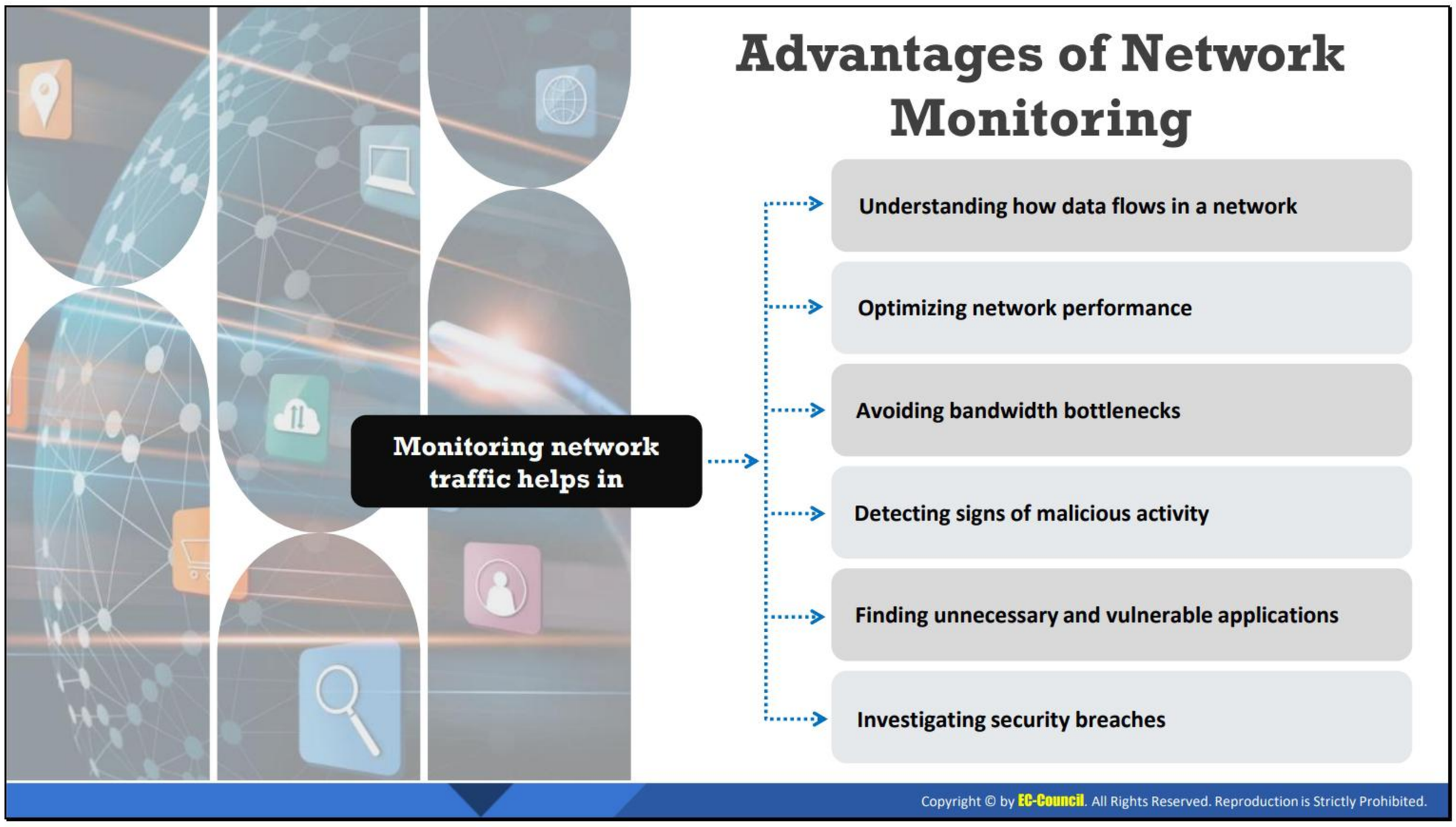
Network operators use network traffic analysis tools to identify malicious or suspicious packets hiding within traffic. They monitor download/upload speeds, throughput, content, traffic behaviors, etc. to understand the status of the network operations.

Need for Network Monitoring

Networking monitoring helps security professionals identify possible issues before they affect business continuity. If an issue occurs in the network, the root cause can be determined easily with network monitoring, and with network automation tools, the problem can be fixed automatically. Networking monitoring not only prevents outages but also gives visibility to potential issues. Continuous network monitoring minimizes downtime and increases the performance of the network.

Even when security tools are in place, attackers can find ways to bypass such security mechanisms to enter the network. Security tools generally use signature-based detection techniques, and it is difficult to identify continuously changing attack signatures/patterns. These tools are not designed to identify behavioral anomalies and are unable to detect attackers' activities that are initiated before and during attacks.

Network monitoring tools provide the first level of security and help identify anomalous conditions in the network, which indicate attacker activity.



Advantages of Network Monitoring

Network traffic analysis is performed to gain in-depth insight into the types of network packets or data flowing through a network. Typically, it is performed through network monitoring or network bandwidth monitoring utilities. The traffic statistics from network traffic analysis helps in the following:

- Understanding how data flows in a network
- Optimizing network performance
- Avoiding bandwidth bottlenecks
- Detecting signs of malicious activity
- Finding unnecessary and vulnerable applications
- Investigating security breaches
- Understanding and evaluating network utilization
- Determining download/upload speeds
- Determining the type, size, origin, destination, and content/data of packets

The typical advantages of network monitoring are as follows.

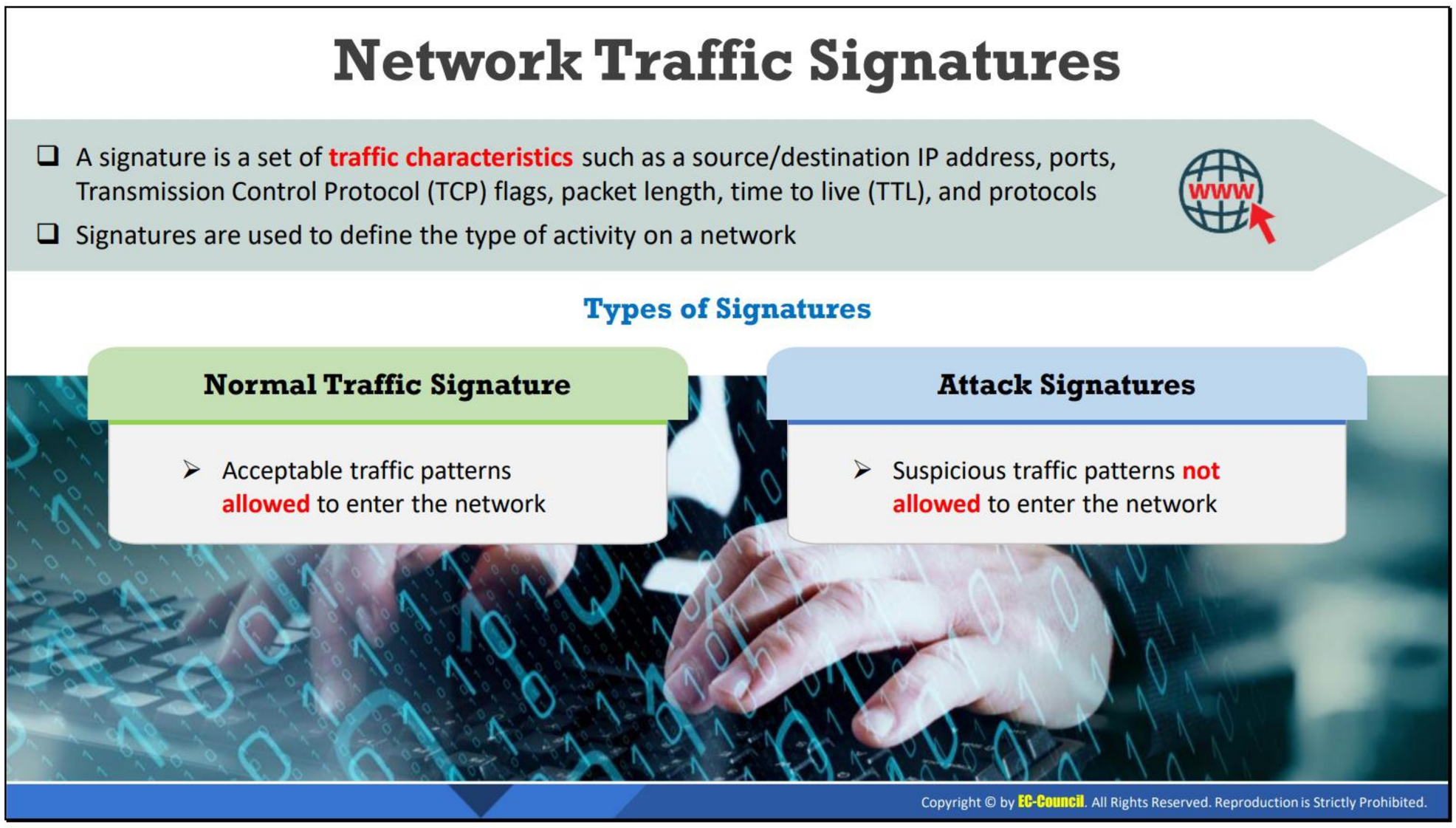
- **Proactive:** Network monitoring proactively detects applications that consume the maximum bandwidth and reduces the bandwidth. It manages server bottleneck situations and other systems connected to the network. Moreover, network monitoring delivers an efficient quality of service to users. It creates a record of all the irregularities occurring in the network that network administrator can handle later.

- **Utilization:** It is important to understand the need for network utilization, especially with all the new and evolving technology. Network monitoring provides complete details on the infrastructure. It provides an idea about the amount of load a network can handle during periods of heavy traffic, enabling the efficient utilization of the space in the network.
- **Optimization:** Network monitoring techniques gather network infrastructure information in a timely manner and save it for the security professionals. Security professional can then take the required actions before the situation worsens. These techniques identify applications that prove vulnerable to the network.
- **Minimizing risk:** Network monitoring techniques are necessary for establishing service-level agreements (SLAs) and compliance applicable to users or consumers. Complete infrastructure information is required when drafting SLAs. The real-time monitoring of network topologies and channels helps in creating the SLAs.

Network monitoring techniques are beneficial for security professionals. They are very easy to setup and implement, considering the complexity of networks.

Determine Baseline Traffic Signatures for Normal and Suspicious Network Traffic

The objective of this section is to explain the various types of network traffic signatures and the concept of baselining normal traffic signatures. It describes the categories of suspicious network traffic signatures and attack signature analysis techniques.



Network Traffic Signatures

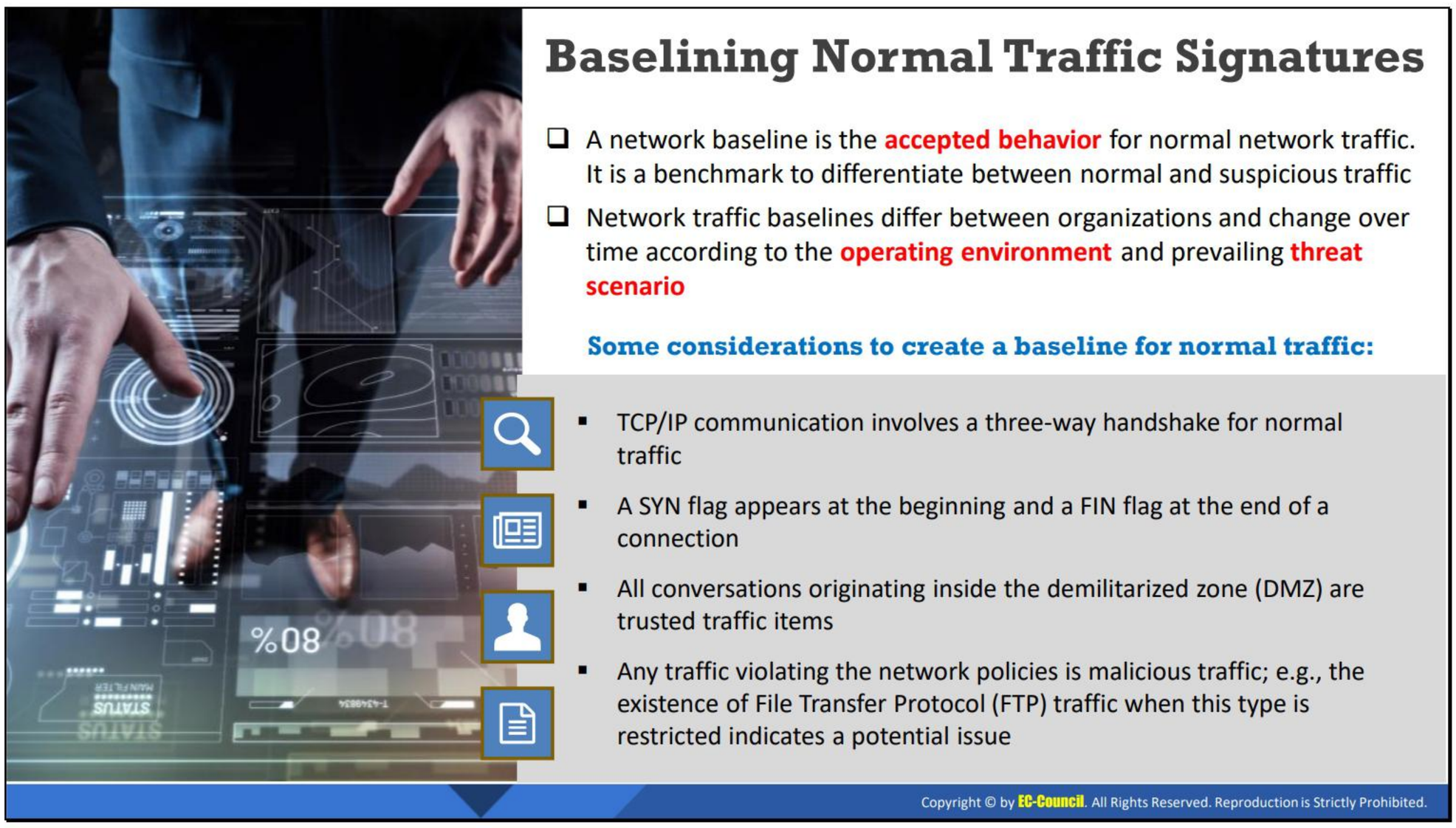
A signature is a set of characters that define network activity, including IP addresses, Transmission Control Protocol (TCP) flags, and port numbers. It includes a set of rules used to detect malicious traffic entering a network. Signatures are used to perform the following:

- Raise alerts in the case of unusual traffic on the network.
- Identify suspicious header characteristics in a packet.
- Configure an intrusion detection system to identify attacks or probes.
- Acquire knowledge on a specific attack that occurred or a vulnerability that can be exploited.
- Match patterns in a packet analysis.

Type of Signatures

Signatures are classified into two main categories depending on their behavior, as described below.

- **Normal traffic signatures**: These include the normal network traffic in the network and are defined based on a normal traffic baseline for the organization. These signatures do not contain any malicious patterns and can be allowed to enter the network.
- **Attack Signatures**: Traffic patterns that appear suspicious are generally treated as attack signatures. These signatures should not be allowed to enter the network. If allowed, they often cause a network security breach. These signatures deviate from the normal signature behavior and should be analyzed.



Baselining Normal Traffic Signatures

A network traffic baseline helps understand the behavioral patterns of a network. It is a benchmark to differentiate between normal and suspicious traffic. Baselining allows a set of metrics to monitor network performance. These metrics define the normal working condition of an enterprise's network traffic. The network traffic is compared with metrics to detect any changes in the traffic that could indicate a security issue in the network. A network traffic baseline establishes the accepted packets that are safe for the organization. Baselining the traffic facilitates the detection of suspicious activities on the network. Any deviation from the normal traffic baseline can be considered a suspicious traffic signature. The security professional should define a network baseline for their organization and validate the traffic against it. Baselining is more effective if it works in parallel with the organization's policy. With the help of normal traffic baselining, security professional can judge the requirements to secure the network. Network traffic baselines differ between organizations and change over time according to the operating environment and prevailing threat scenario.

Although, there is no industry standard to measure network traffic performance baselines, there are network monitoring tools that provide estimates of what type of traffic is normal. A network traffic baseline should be defined for all incoming, and outgoing Internet traffic and wide area network (WAN) links. The network traffic baseline should also contain the traffic for critical business data and backup systems.

- According to a network traffic baseline, normal traffic signatures for TCP packets should have the following characteristics:
  - To establish a three-way handshake, TCP uses SYN, SYN ACK, and ACK bits in every session.

- The ACK bit should be set in every packet, except for the initial packet, in which the SYN bit is set.
- FIN ACK and ACK are used in terminating a connection. PSH FIN and ACK may also be used initially in the same process.
- RST and RST ACK are used to quickly end an on-going connection.
- During a conversation (after a handshake and before termination), packets only contain an ACK bit by default. Occasionally, they may also have a PSH or URG bit set.

- A suspicious TCP packet has one or more of the following characteristics:
  - If both SYN and FIN bits are set, the TCP packet is illegal.
  - SYN FIN PSH, SYN FIN RST, and SIN FIN PSH RST are all variants of SIN FIN. An attacker sets these additional bits to avoid detection.
  - A packet having only a FIN flag is illegal as FIN can be used in network mapping, port scanning, and other stealth activities.
  - Some packets have all six flags unset; these are known as NULL flags and are illegal.
  - The source or destination port is zero.
  - If the ACK flag is set, then the acknowledgement number should not be zero.
  - If a packet has only the SYN bit, which is set at the beginning to establish a connection, and any other data are present, then it is an illegal packet.
  - If the destination address is a broadcast address (ending with 0 or 255), it is an illegal packet.
  - Every TCP packet has two bits reserved for future use. If either or both are set, then the packet is illegal.
- All conversations originating inside the demilitarized zone (DMZ) are trusted traffic items.
- Any traffic violating the network policies is malicious traffic, e.g., the existence of File Transfer Protocol (FTP) traffic when this type is restricted indicates a potential issue.
- Any Dynamic Host Configuration Protocol (DHCP) traffic from unknown DHCP servers indicates a rogue DHCP server.
- Mail traffic originating in the network but not sent to a mail server is suspect.
- Any DNS traffic not sent to the DNS server is suspect.
- Any outgoing traffic with internal addresses not matching the organization's address space may be malicious.

Categories of Suspicious Traffic Signatures

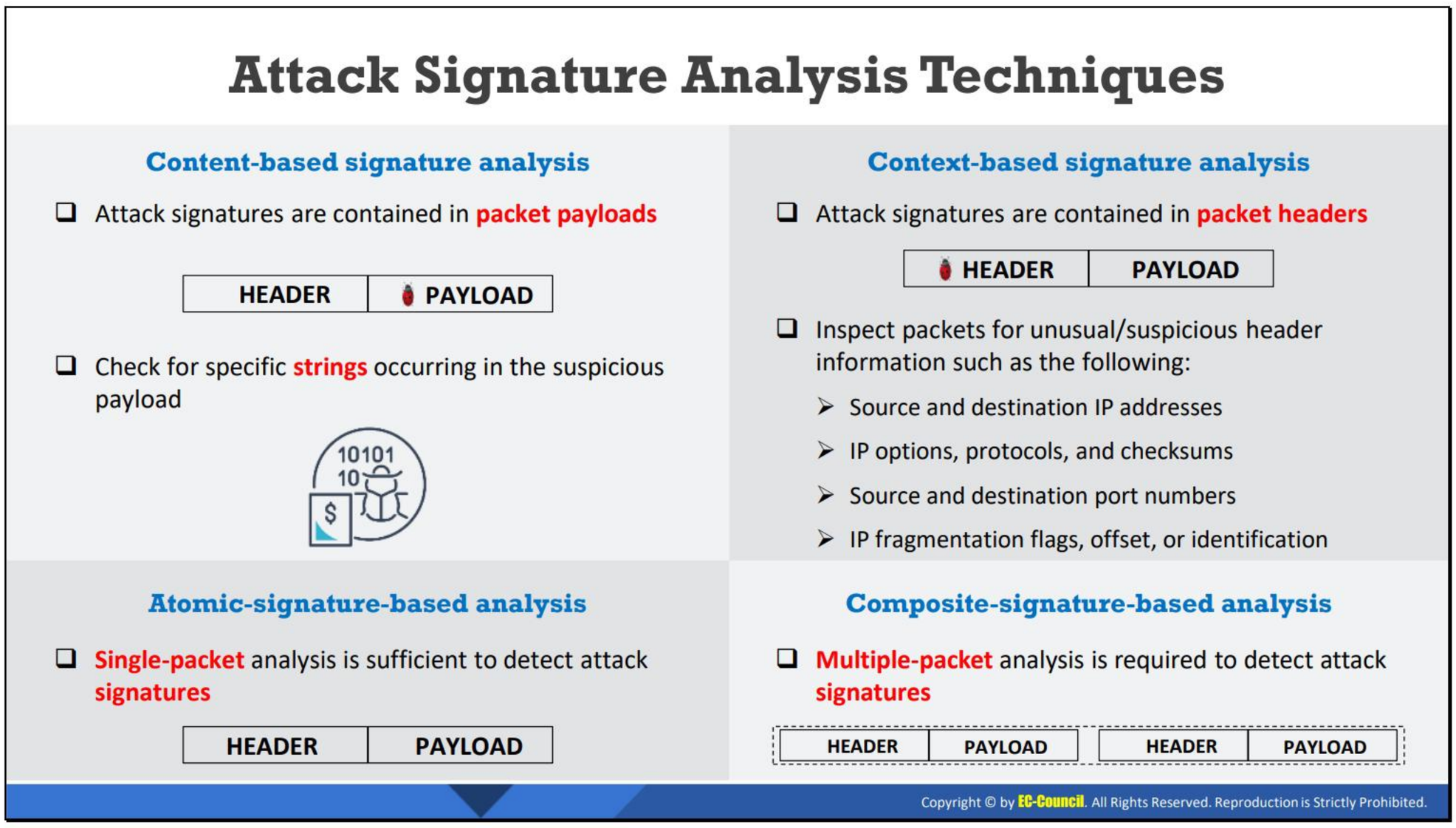
Network traffic deviating from normal behavior is categorized as a suspicious traffic signature. It is classified into four categories as follows.

- **Informational:** The informational traffic signature detects normal network activity. Although it may not appear suspicious, the data gathered through the informational signature can be used for suspicious activities. For example, informational traffic signatures may include the following:
  - Internet Control Message Protocol (ICMP) echo requests
  - TCP connection requests
  - User Datagram Protocol (UDP) connections
- **Reconnaissance**: Reconnaissance traffic consists of signatures that indicate an attempt to scan the network for possible weaknesses. Reconnaissance is an unauthorized discovery of vulnerabilities, which maps of systems and services. Reconnaissance is also known as information gathering, and it precedes a network attack in most cases. For example, reconnaissance traffic signatures may include the following:
  - Ping sweep attempts
  - Port scan attempts
  - Domain Name System (DNS) query attempts
- **Unauthorized access:** Traffic may contain signs of someone attempting to gain unauthorized access, unauthorized data retrieval, system access or privilege escalation, etc. An attacker who does not have privileges to access an organization's network

usually generates this type of traffic with the intention of capturing sensitive data. For example, unauthorized access traffic signatures may include the following:

- Password cracking attempts
- Sniffing attempts
- Brute-force attempts

- **Denial of service (DoS):** This type of traffic may contain a large number of requests from a single source or multiple sources, which are sent as an attempt to perform a DoS attack. This type of attack is performed to disrupt the service of the target organization. For example, DoS traffic signatures may include the following:
  - Ping of death attempts
  - SYN flood attempts



Attack Signature Analysis Techniques

Attack signature analysis techniques are classified into four different categories as follows.

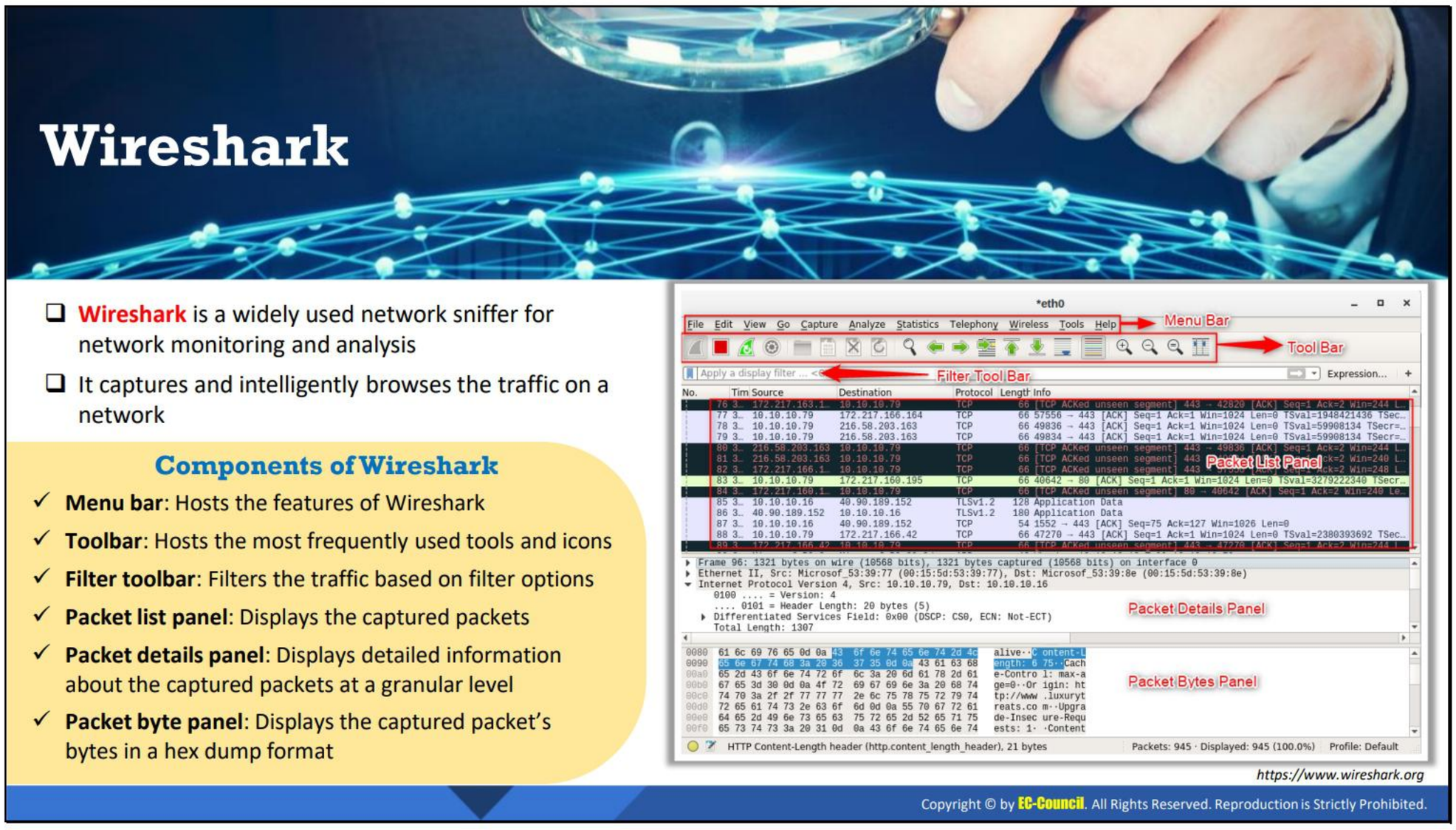
- **Content-based signature analysis:** Content-based signatures are detected by analyzing the data in the payload and matching a text string to a specific set of characters. If undetected, these signatures can open backdoors in a system, providing administrative controls to an outsider.
- **Context-based signature analysis:** Packets are usually altered using the header information. Suspicious signatures in the header can include malicious data that can affect the following:
  - Source and destination IP addresses
  - Source and destination port numbers
  - IP options
  - IP protocols
  - IP, TCP, and UDP checksums
  - IP fragmentation flags, offset, or identification
- **Atomic-signature-based analysis:** To detect an atomic signature, security professionals need to analyze a single packet to determine whether the signature includes malicious patterns. Security professionals do not require any knowledge of past or future activities to detect these signature patterns.
- **Composite-signature-based analysis:** In contrast to atomic signatures, security professionals need to analyze a series of packets over a long period of time to detect

composite attack signatures. Detecting these attack patterns is exceedingly difficult. ICMP flooding is an example of an attack performed using composite signatures. In this attack, multiple ICMP packets are sent to a single host so that the server remains busy responding to the requests.

Attacker signatures may be located in either the header or payload of the packet.

Perform Network Monitoring for Suspicious Traffic

The objective of this section is to explain how to use Wireshark to perform network monitoring and analysis. It describes how to use Wireshark for monitoring and analyzing File Transfer Protocol (FTP) traffic, Telnet traffic, and Hypertext Transfer Protocol (HTTP) traffic.



Wireshark

Source: *https://www.wireshark.org*

Wireshark is a packet sniffer that can be used for network troubleshooting to investigate security issues and to analyze and understand network protocols. It can exploit information passed in plain text.

- **Features**

  Wireshark has a rich feature set that includes the following:

  - Identify poor network performance due to high path latency.
  - Locate internetwork devices that drop packets.
  - Validate the optimal configuration of network hosts.
  - Analyze application functionality and dependencies.
  - Optimize application behavior for best performance.
  - Analyze network capacity before application launch.
  - Verify application security during launch, login, and data transfer.
  - Identify unusual network traffic indicating potentially compromised hosts.

- **Prerequisites for network packet capture**

  Setting up Wireshark to capture packets for the first time can be tricky. The following are a few common problems that are encountered while capturing packets with Wireshark for the first time:

  - Special privileges are required to start a live capture.

- The correct network interface must be chosen to capture packet data from.
- Network packets should be captured at the correct location in the network to view the desired traffic.

- **Wireshark network analysis activities**

  Capturing live network data is one of the major features of Wireshark. The Wireshark capture engine enables security professionals to perform the following:

  - Capture from different types of network hardware such as Ethernet and 802.11.
  - Stop the capture based on different triggers such as the amount of captured data, elapsed time, or number of packets.
  - Simultaneously show decoded packets while capturing is in progress.
  - Filter packets to reduce the amount of data to be captured.
  - Save packets in multiple files during a long capture.
  - Simultaneously capture from multiple network interfaces.

- **First network packet capture using Wireshark**

  To capture packets using Wireshark, first install and launch the tool on the target network. Select the appropriate network interface to capture traffic from. The following are the steps to start capturing packets with Wireshark:

  1. Double-click on an interface in the main window.
  2. An overview of the available interfaces can be obtained using the Capture Interface dialog box.
  3. Start a capture from this dialog box using the Start button.
  4. A capture can be immediately started using the current settings by selecting **Capture → Start** or by clicking the first toolbar button.
  5. If the name of the capture interface is known, Wireshark can be launched from the command line through the following command: $ wireshark -i eth0 –k

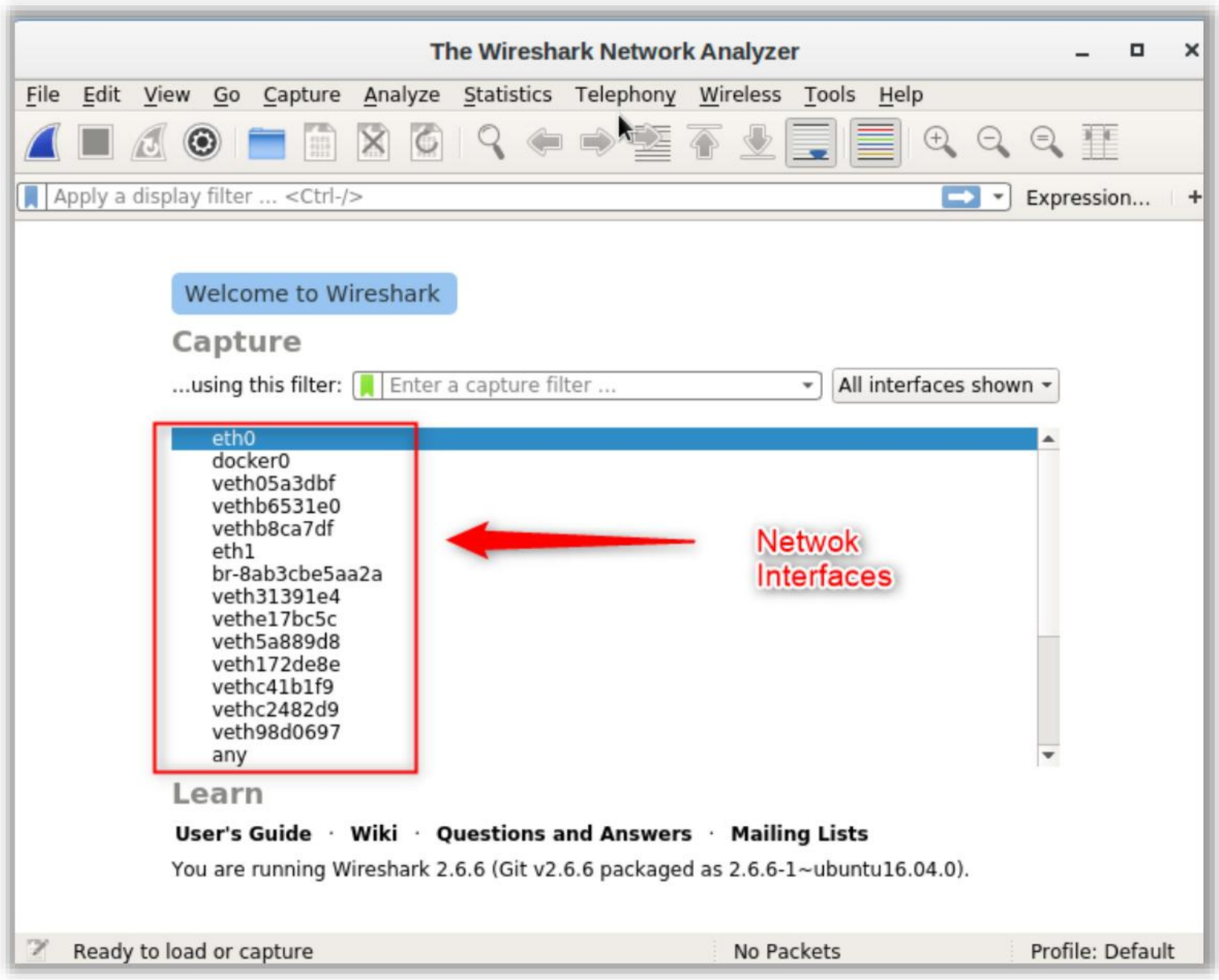


Figure 17.1: Wireshark Network Interfaces

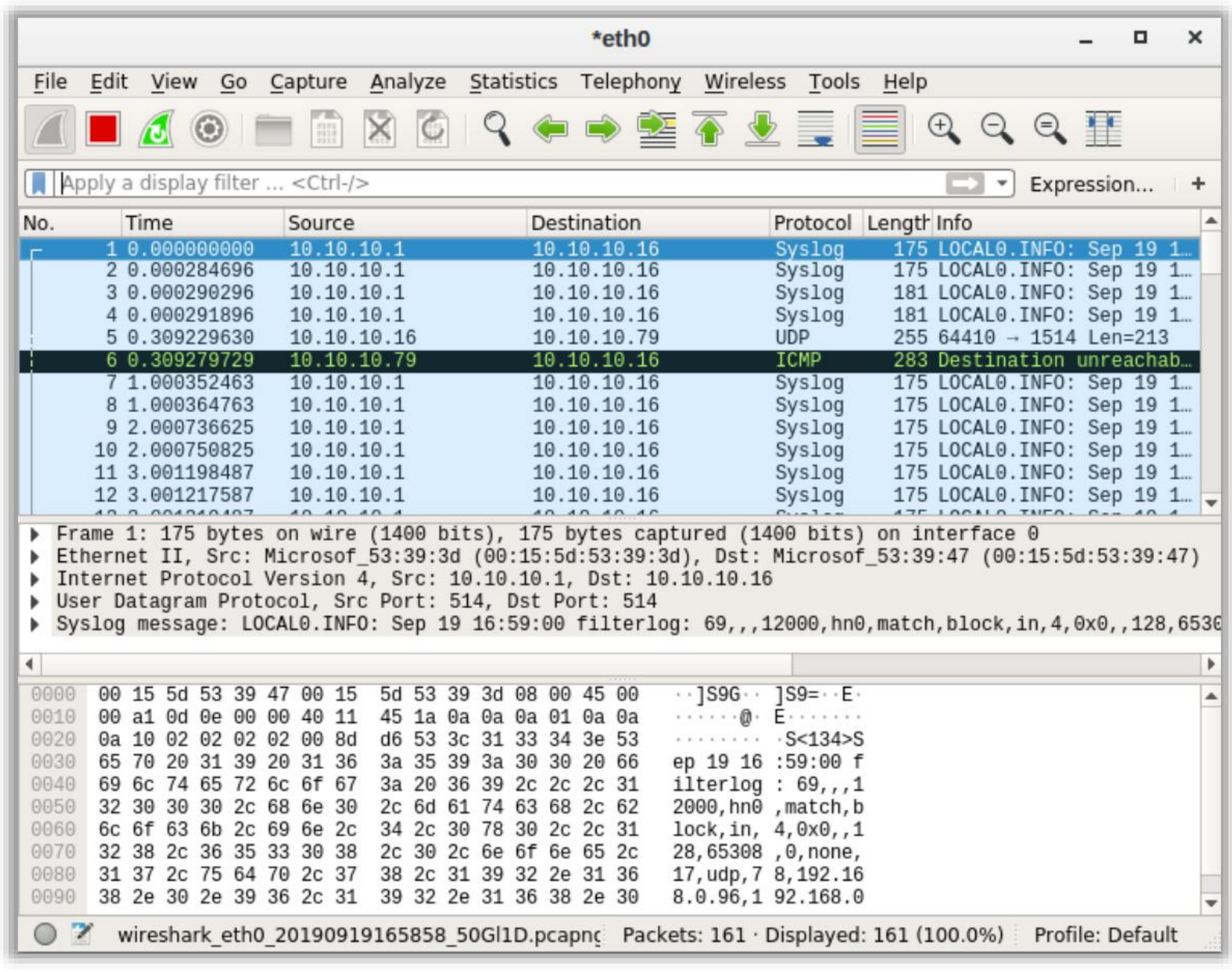


Figure 17.2: Capturing Traffic

Wireshark components

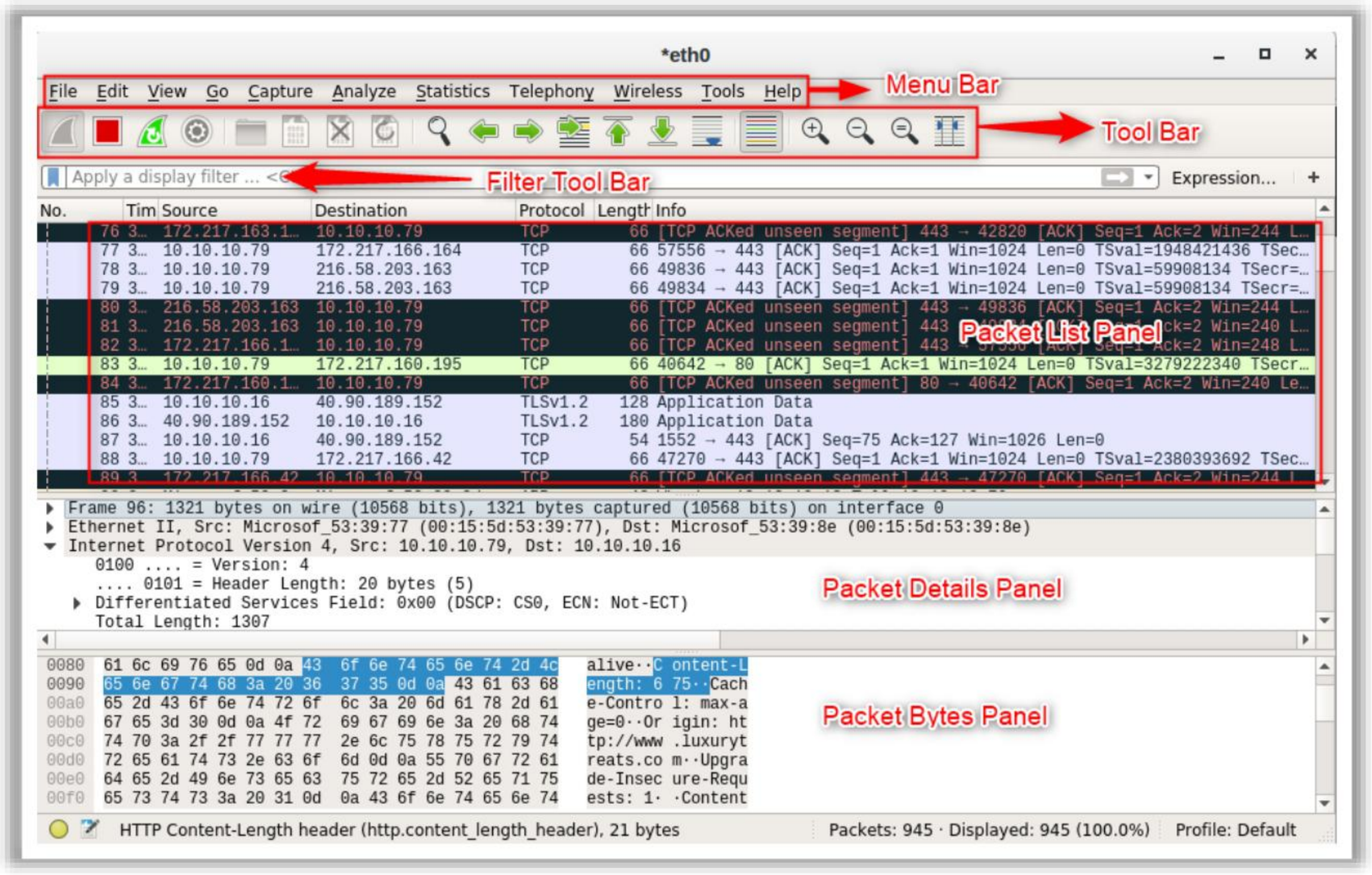


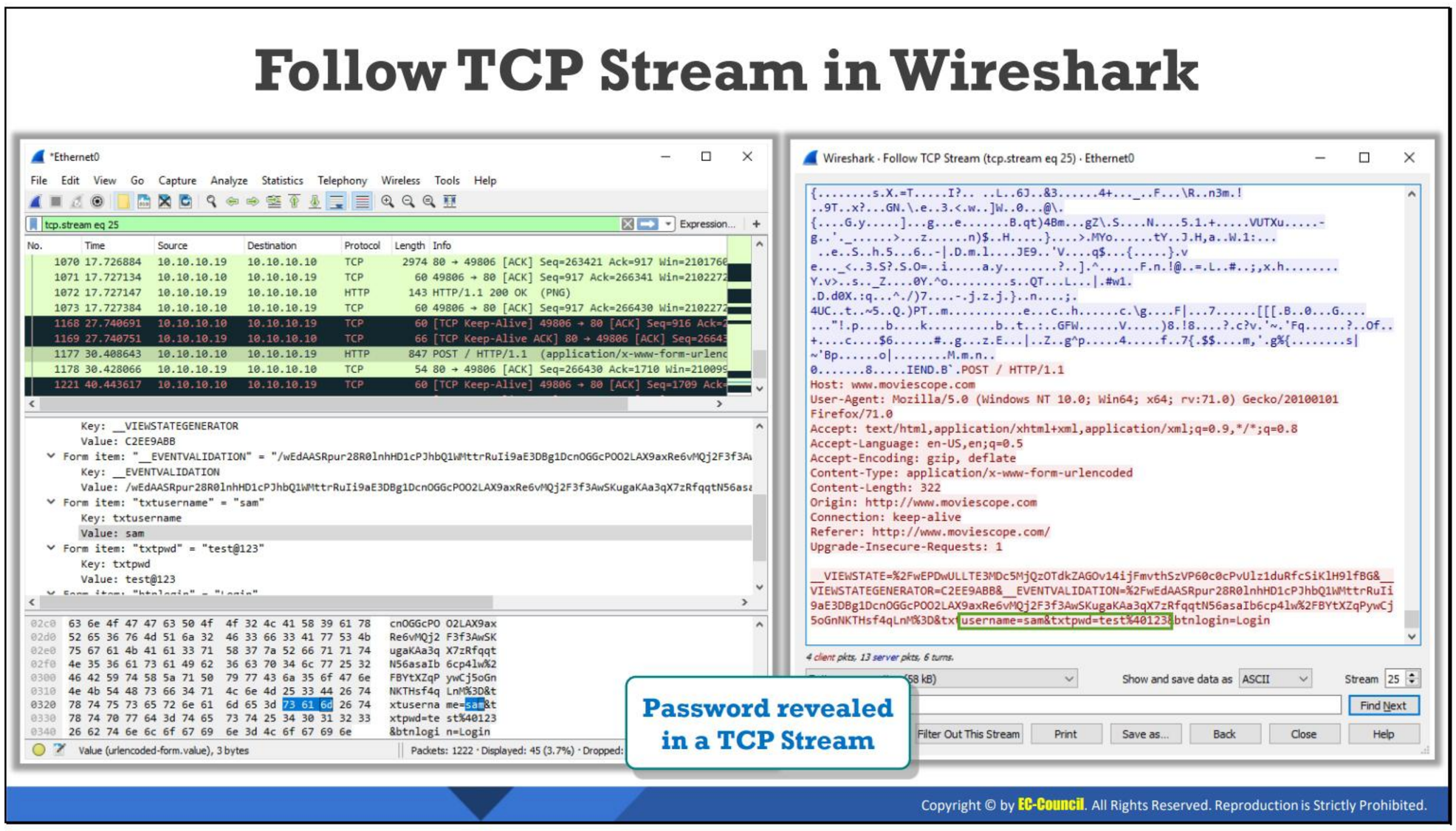
Figure 17.3: Wireshark components

The main menu of Wireshark contains the following items:

- **File:** This menu contains items to open and merge, capture files, save, print, import and export capture files in whole or in part, and quit the Wireshark application.
- **Edit:** This menu contains items to find a packet, time reference, and mark one or more packets. It handles configuration profiles and sets preferences.
- **View:** This menu controls the display of the captured data, including the colorization of packets, font zoom, display of a packet in a separate window, and expanding and collapsing of the packet tree details.
  - **Colorize packet list:** This option allows security professionals to control whether Wireshark should colorize the packet list. Enabling colorization slows down the display of new packets while capturing and loading capture files.
  - **Coloring rules:** This option allows security professionals to color packets in the packet list pane according to the filter expressions of their choice. It can be very useful for spotting certain types of packets.
  - **Colorize conversation:** This menu item brings up a submenu that allows the color of the packets to be changed in the packet list pane based on the addresses of the currently selected packet. This makes it easy to distinguish packets belonging to different conversations.
- **Go:** This menu contains options to navigate to a specific packet including a previous packet, the next packet, the corresponding packet, the first packet, and the last packet.

- **Capture:** This menu allows the security professionals to start, stop, and restart capture and to edit capture filters.
  - **Capture filters:** This option allows security professionals to create and edit capture filters. Filters can be named and saved for future use.
- **Analyze:** This menu contains items to manipulate, display and apply filters, enable or disable the dissection of protocols, configure user-specified decodes, and follow a different stream including TCP, UDP, and Secure Sockets Layer (SSL).
  - **Follow TCP stream:** This option displays all the captured TCP segments that are on the same TCP connection as a selected packet.
  - **Follow UDP stream:** This option displays all the captured UDP segments that are on the same UDP connection as a selected packet.
  - **Follow SSL stream:** This option displays all the captured SSL segments that are on the same SSL connection as a selected packet.
- **Statistics:** This menu contains options to display various statistic windows, including a summary of the packets that have been captured, display protocol hierarchy statistics, IO graphs, and flow graphs.
- **Telephony:** This menu contains options to display various telephony-related statistic windows including a media analysis, flow diagrams, and display protocol hierarchy statistics.
- **Wireless:** This menu shows Bluetooth and IEEE 802.11 wireless statistics.
- **Tools:** This menu contains various tools available in Wireshark including the creation of firewall access control list (ACL) rules and use of the Lua interpreter.
  - **Firewall ACL rules:** This tool can be used create command-line ACL rules for many different firewall products, including Cisco IOS, Linux Netfilter, OpenBSD and Windows Firewall. Rules for MAC addresses, IPv4 addresses, TCP and UDP ports, and IPv4+port combinations are supported. It is assumed that the rules will be applied to an outside interface.
  - **Lua:** This tool includes options that allow security professionals to work with the built-in Lua interpreter of Wireshark. Wireshark uses Lua to write protocol dissectors.
- **Help:** This menu contains items to help the user, including access to basic help manual pages for the various command-line tools, online access to some webpages, and the About Wireshark dialog.
- **Main toolbar:** The main toolbar provides quick access to frequently used items from the menu. This toolbar cannot be customized by the user. If the space on the screen is needed to show more packet data, then the toolbar can be hidden using the View menu. As in the menu, only the items that can be used in the current program state will be available. The others will be greyed out.

- **Filter toolbar:** The filter toolbar allows security professionals to quickly edit and apply display filters.
- **Packet list panel:** This panel displays a list of packets in the current capture file. It colors the packets based on the protocol. Each line in the packet list corresponds to one packet in the capture file. If a line in this pane is selected, more details will be displayed in the Packet Details and Packet Bytes panes.
- The default columns show the following:
  - **No:** This column shows the number of the packets in the capture file. This number does not change, even if a display filter is used.
  - **Time**: This column shows the timestamp of the packet. The presentation format of this timestamp can be changed.
  - **Source:** This column shows the source address of the packet.
  - **Destination**: This column shows the destination address of the packet.
  - **Protocol**: This column shows the protocol name in the abbreviated form.
  - **Info**: This column shows additional information about the packet content.
- **Packet details panel:** This panel displays the details of the selected packet. It includes the different protocols making up the layers of data in this packet. The protocols and fields of the packet are displayed using a tree, which can be expanded and collapsed. Layers include the frame, Ethernet, IP, TCP, UDP, ICMP, and application protocols such as HTTP.
- **Packet bytes panel:** This panel displays the packet bytes in a hex dump and American Standard Code for Information Interchange (ASCII) encodings. For a hex dump, the left side shows the offset in the packet data, and the middle of the packet data is shown in a hexadecimal representation. On the right, the corresponding ASCII characters are displayed.
- **Status bar:** The status bar displays informational messages. In general, the left side shows context-related information, the middle part shows the current number of packets, and the right side shows the selected configuration profile. The user can drag the handles between the text areas to change the size.



Follow TCP Stream in Wireshark

Source: *https://www.wireshark.org*

Wireshark displays data from the TCP port with a feature known as "**Follow TCP stream**." The tool sees TCP data in the same way as that of the application layer. Use this tool to find passwords in a telnet session or to interpret a data stream.

To see the TCP stream, select a TCP packet in the packet list of a stream/connection and then select the **Follow TCP Stream** menu item from the Wireshark **Tools** menu. Wireshark displays all the data from the TCP stream by setting an appropriate display filter. The tool displays the streaming content in the same sequence as it appeared on the network. It displays the captured data in ASCII, EBCDIC, hex dump, C array, or raw formats.

As shown in the screenshot, you can capture network traffic and gain the credentials of a target machine. You can attempt to capture its remote interface and monitor the traffic generated from a user's browsing activities to extract confidential user information.

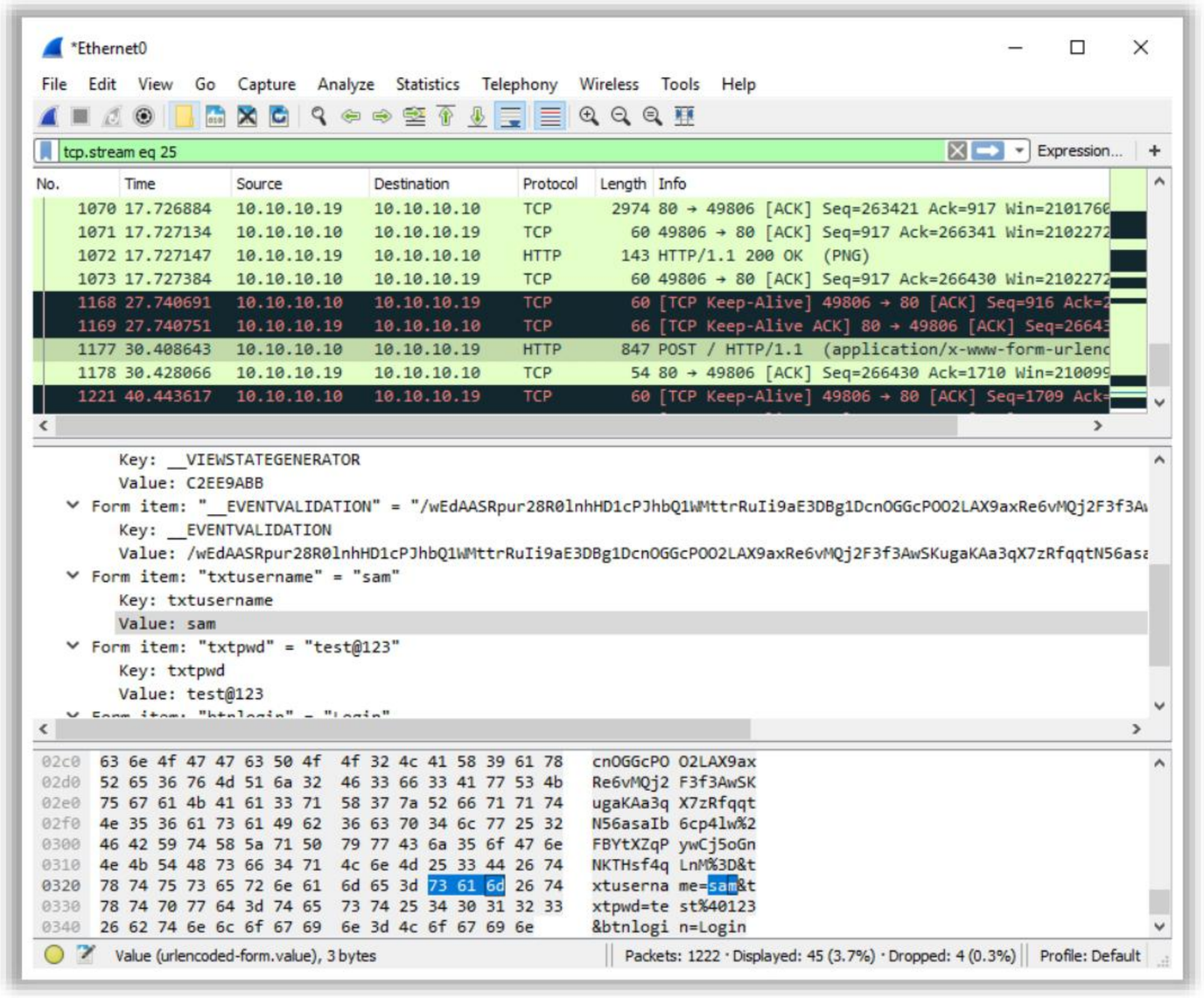


Figure 17.4: Wireshark capturing TCP Stream

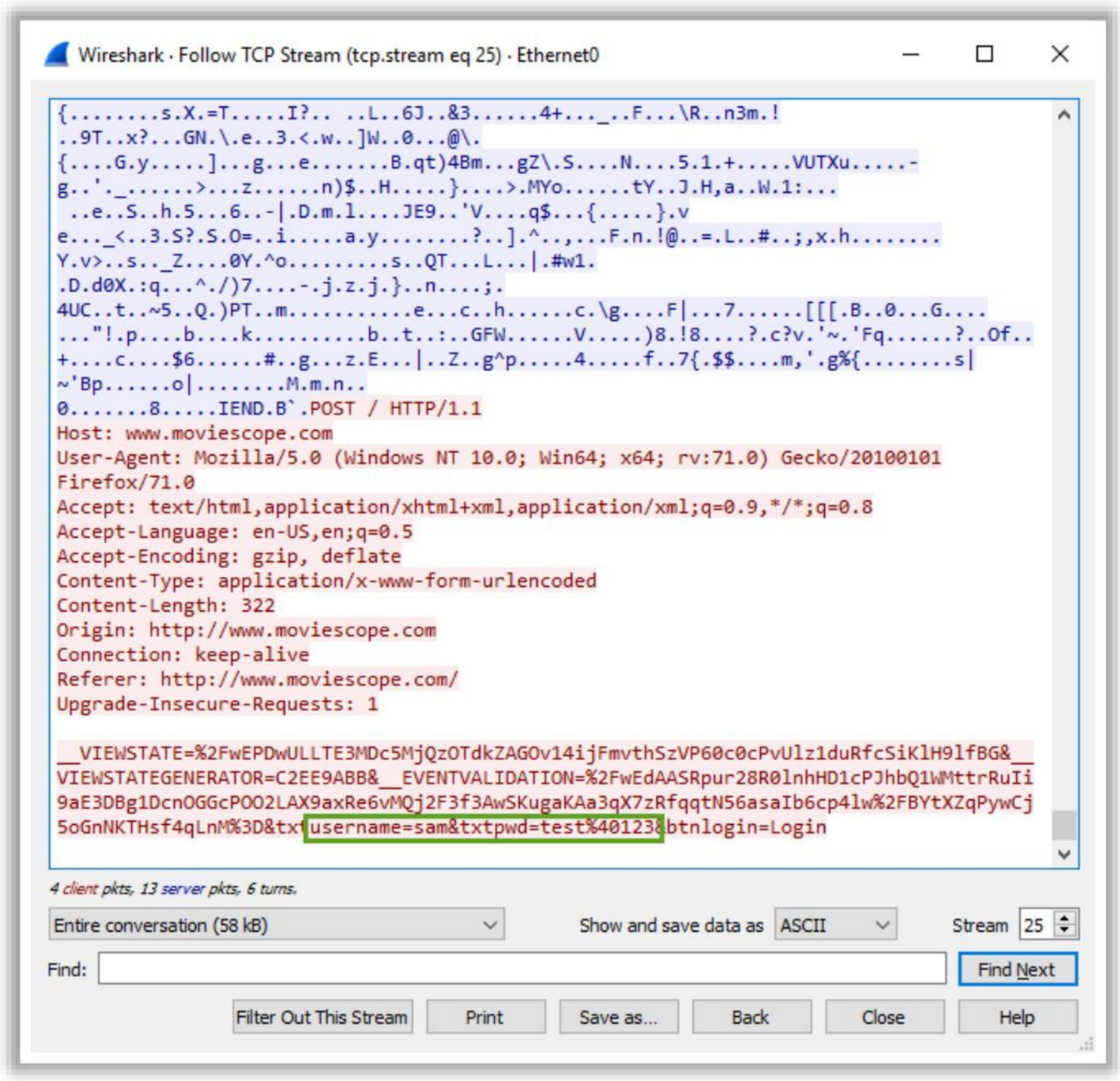
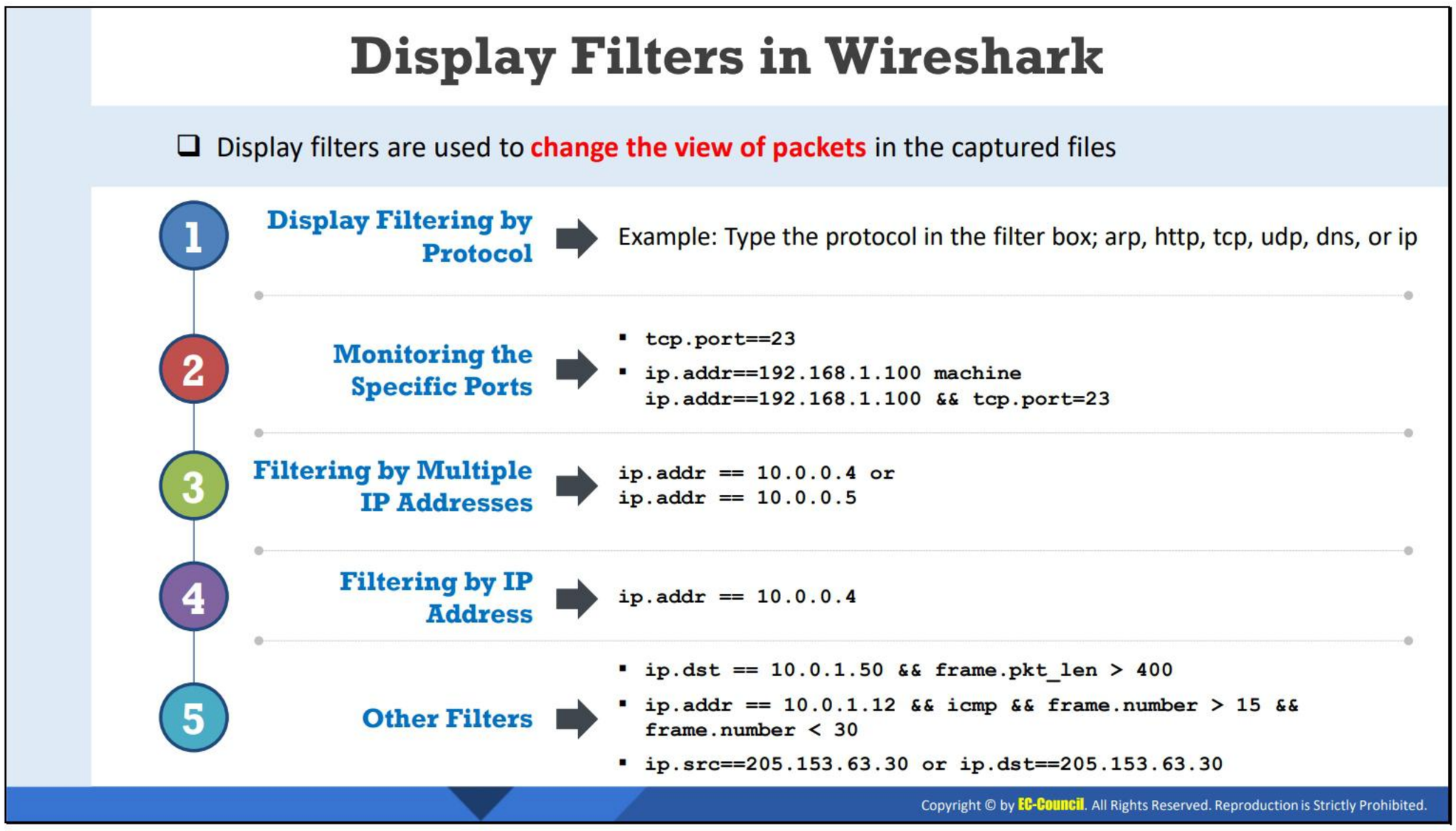


Figure 17.5: Wireshark feature Follow TCP Stream



Display Filters in Wireshark

Source: *https://wiki.wireshark.org*

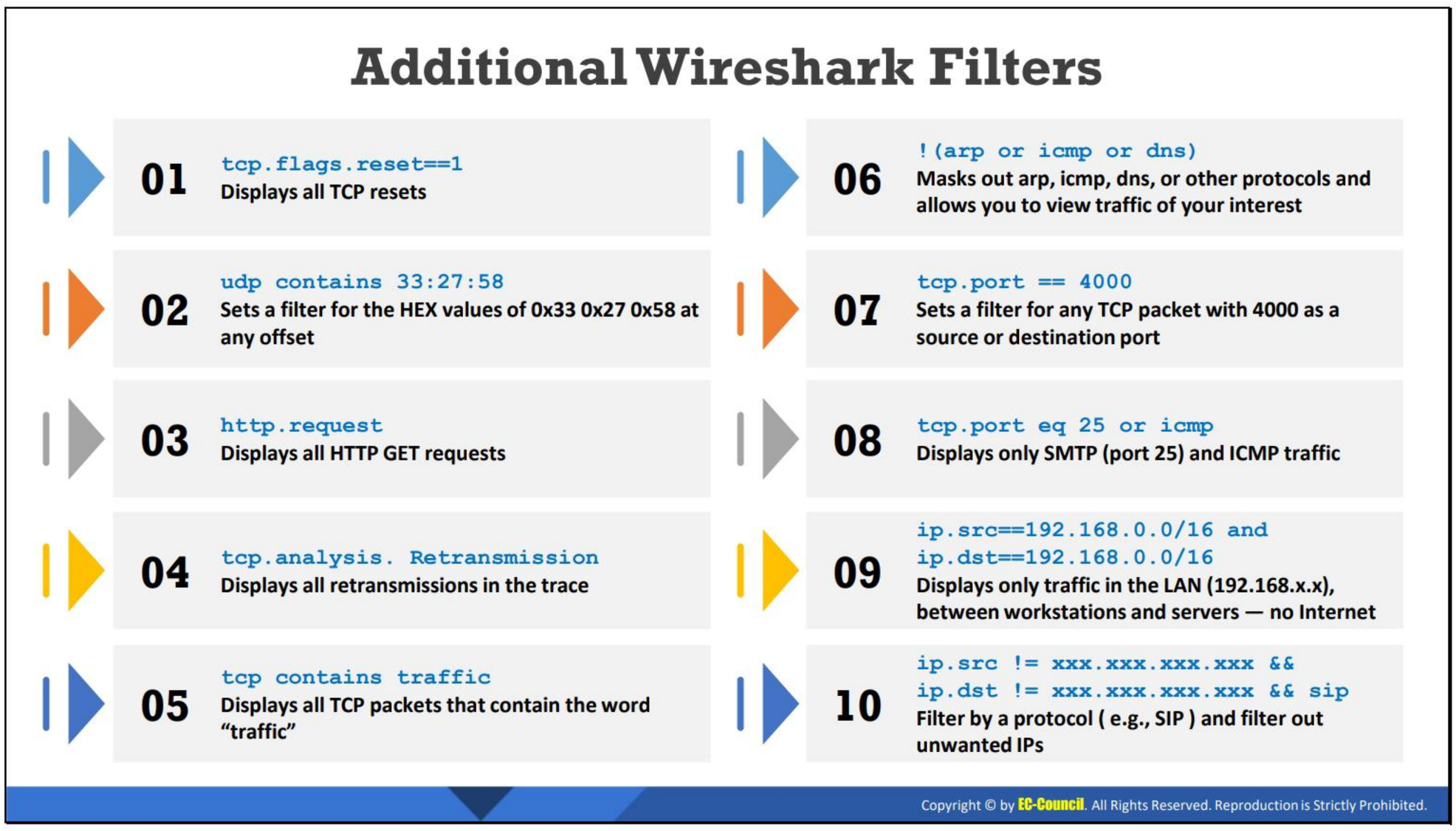
Wireshark features display filters that filter traffic on the target network by protocol type, IP address, port, etc. Display filters are used to change the view of packets in the captured files. To set up a filter, type the protocol name, such as arp, http, tcp, udp, dns, and ip, in the filter box of Wireshark. Wireshark can use multiple filters at a time.

Some of the display filters in Wireshark are listed below:

- **Display Filtering by Protocol**

  Example: Type the protocol in the filter box: arp, http, tcp, udp, dns, ip

- **Monitoring the Specific Ports**
  - `tcp.port==23`
  - `ip.addr==192.168.1.100 machine`
    `ip.addr==192.168.1.100 && tcp.port==23`
- **Filtering by Multiple IP Addresses**
  - `ip.addr == 10.0.0.4 or ip.addr == 10.0.0.5`
- **Filtering by IP Address**
  - `ip.addr == 10.0.0.4`
- **Other Filters**
  - `ip.dst == 10.0.1.50 && frame.pkt_len > 400`
  - `ip.addr == 10.0.1.12 && icmp && frame.number > 15 && frame.number < 30`
  - `ip.src==205.153.63.30 or ip.dst==205.153.63.30`



Additional Wireshark Filters

Source: *https://wiki.wireshark.org*

Some examples of additional Wireshark filters are listed below:

- **`tcp.flags.reset==1`**

  Displays all TCP resets

- **`udp contains 33:27:58`**

  Sets a filter for the hex values of 0×33 0×27 0×58 at any offset

- **`http.request`**

  Displays all HTTP GET requests

- **`tcp.analysis.retransmission`**

  Displays all retransmissions in the trace

- **`tcp contains traffic`**

  Displays all TCP packets that contain the word "traffic"

- **`!(arp or icmp or dns)`**

  Masks out arp, icmp, dns, or other protocols and allows you to view the traffic of your interest

- **`tcp.port == 4000`**

  Sets a filter for any TCP packet with 4000 as a source or destination port

- **`tcp.port eq 25 or icmp`**

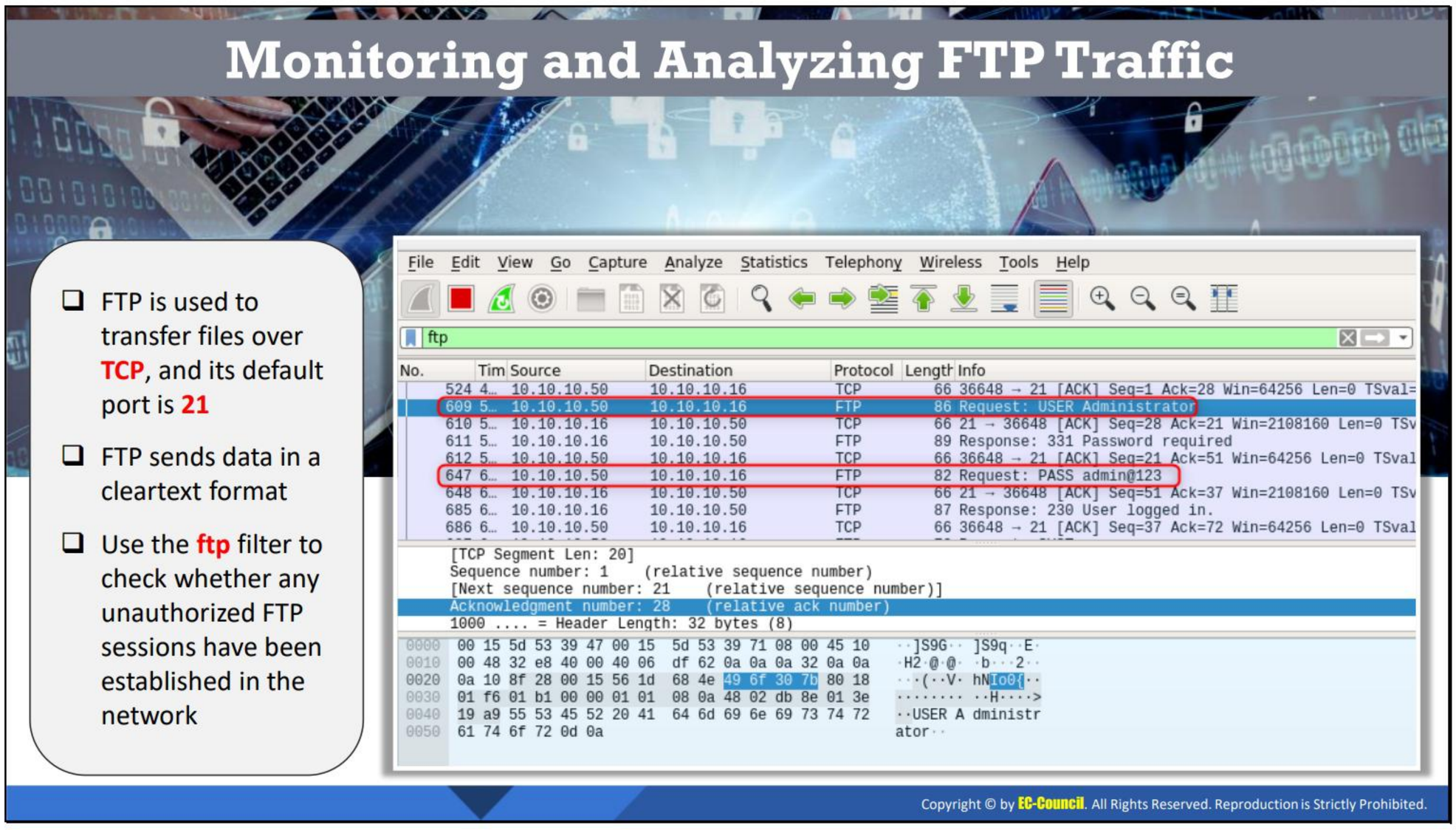
  Displays only SMTP (port 25) and ICMP traffic

- **`ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16`**

  Displays only traffic in the LAN (192.168.x.x), between workstations and servers—no Internet

- **`ip.src != xxx.xxx.xxx.xxx && ip.dst != xxx.xxx.xxx.xxx && sip`**

  Filters by a protocol (e.g., SIP) and filters out unwanted IPs



Monitoring and Analyzing FTP Traffic

FTP offers neither a secure network environment nor secure user authentication. Individuals do not need authentication to access an FTP server in a network. This provides an easy method for attackers to enter the network and access resources. FTP does not provide encryption in the data transfer process, and the data transfer between the sender and receiver is in plain text. Consequently, critical information such as usernames and passwords are exposed to attackers. The implementation of FTP in an organization's network leaves the data accessible to external sources. Deploying FTP in a network can lead to types of attacks such as FTP bounce, FTP brute force, and packet sniffing attacks.

Wireshark provides complete information about the FTP traffic on a network for monitoring. Applying an FTP filter helps detect unauthorized sessions running on the server. Apart from monitoring the traffic on the FTP server, the existing file contents and file sizes in the server should be monitored.

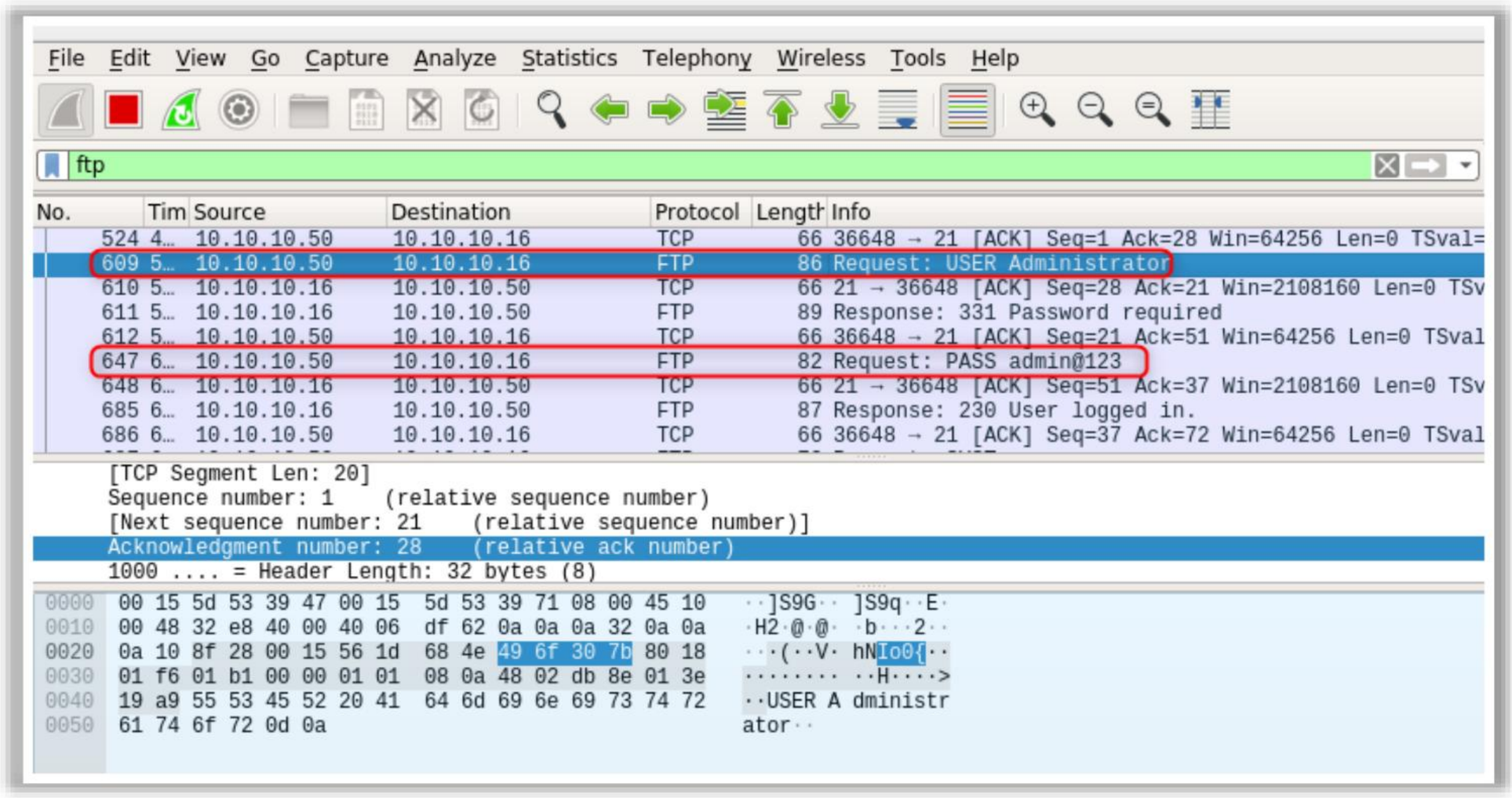


Figure 17.6: FTP traffic

Monitoring and Analyzing Telnet Traffic

Telnet can provide access to remote hosts including most network equipment and operating systems

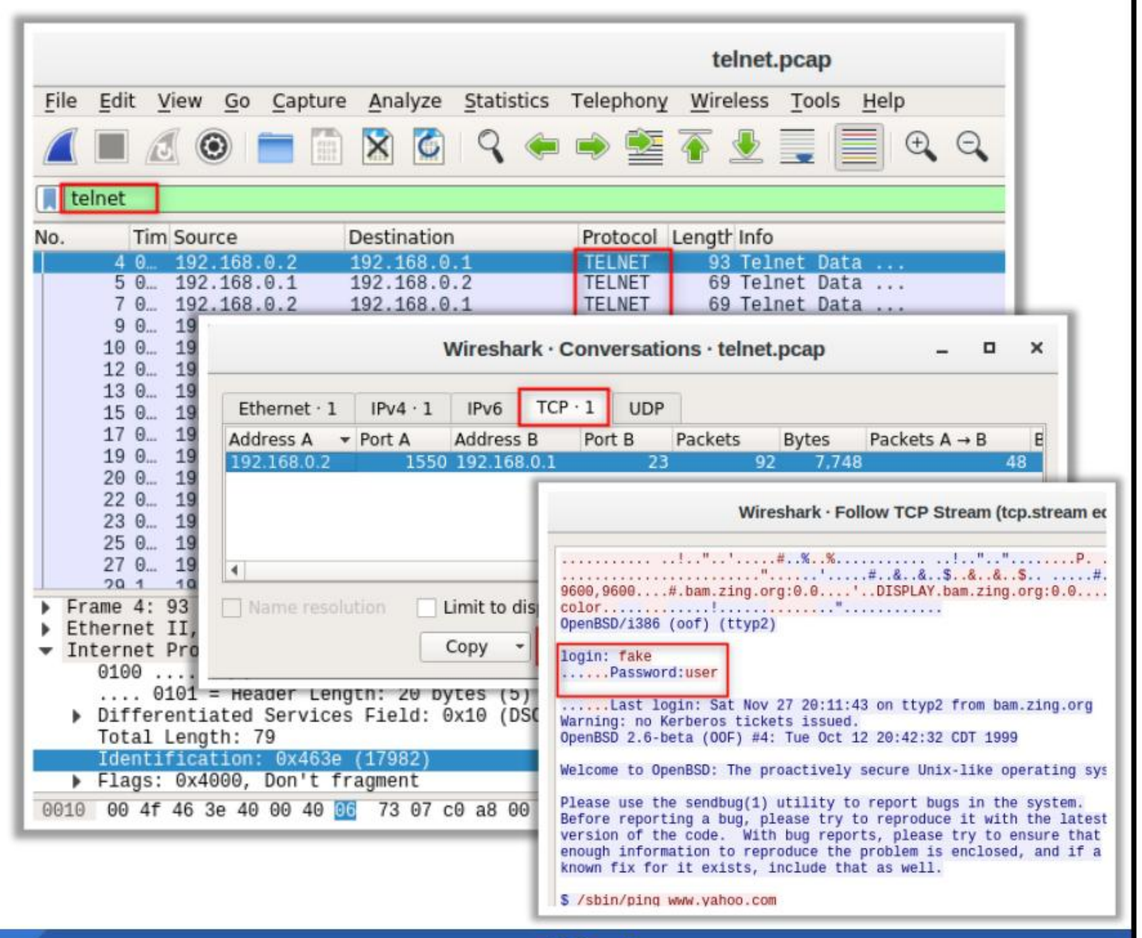
Telnet is **not encrypted**; the password and all other data are transmitted as cleartext

Ideally, it should be disabled; enabling it poses huge **security risks** to the network

It is essential to check whether any Telnet session is established within the network

Monitoring and Analyzing Telnet Traffic (Cont'd)

- Perform the following to check for established Telnet sessions:
  - Go to the **Statistics** Menu and click on **Conversations**
  - Go to the **TCP tab** and select the appropriate Telnet communication indicated by port 23 and click **Follow Stream...**
  - The Telnet traffic and the credentials will be viewable in cleartext



Monitoring and Analyzing Telnet Traffic

The Telnet protocol works on a client–server model. It provides access to remote network equipment and OSes. The data transferred through Telnet is not encrypted, making it easy for intruders to eavesdrop. If a person has access to a network device with Telnet configured, they can gain access to the network and user account information. Generally, Telnet should be disabled in an organization.

Telnet is a session-oriented protocol, which implies that the connection must be open for the entire session. Attackers can use Telnet open sessions to perform a network security breach. Therefore, security professionals should monitor Telnet sessions (if any) running on their network. Timely monitoring of Telnet sessions through Wireshark can greatly minimize the risk for network intrusion.

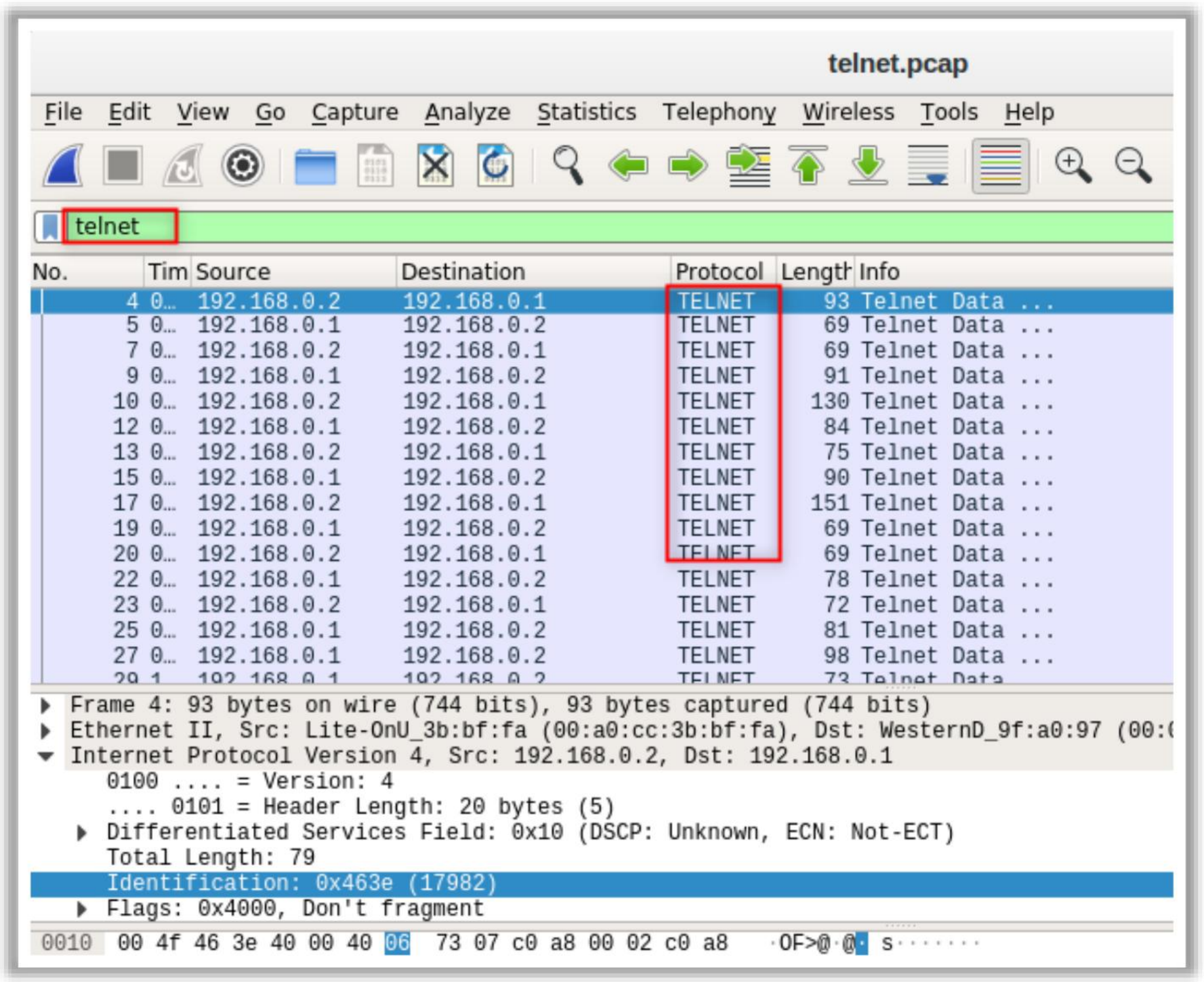


Figure 17.7: Telnet traffic

- Perform the following to check for established Telnet sessions:
  - Go to the **Statistics** Menu and click on **Conversations.**
  - Go to the **TCP tab** and select the appropriate Telnet communication indicated by port 23 and click **Follow Stream…**

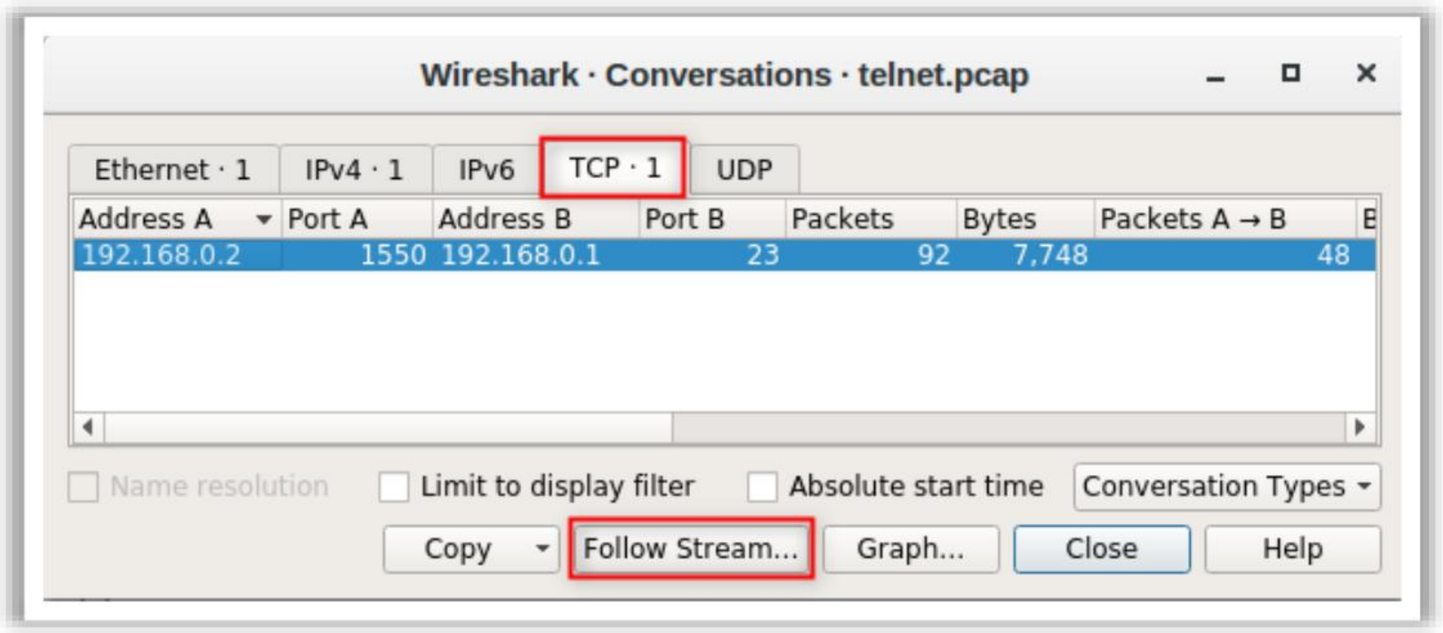


Figure 17.8: TCP tab

- The Telnet traffic and the credentials will be viewable in cleartext.

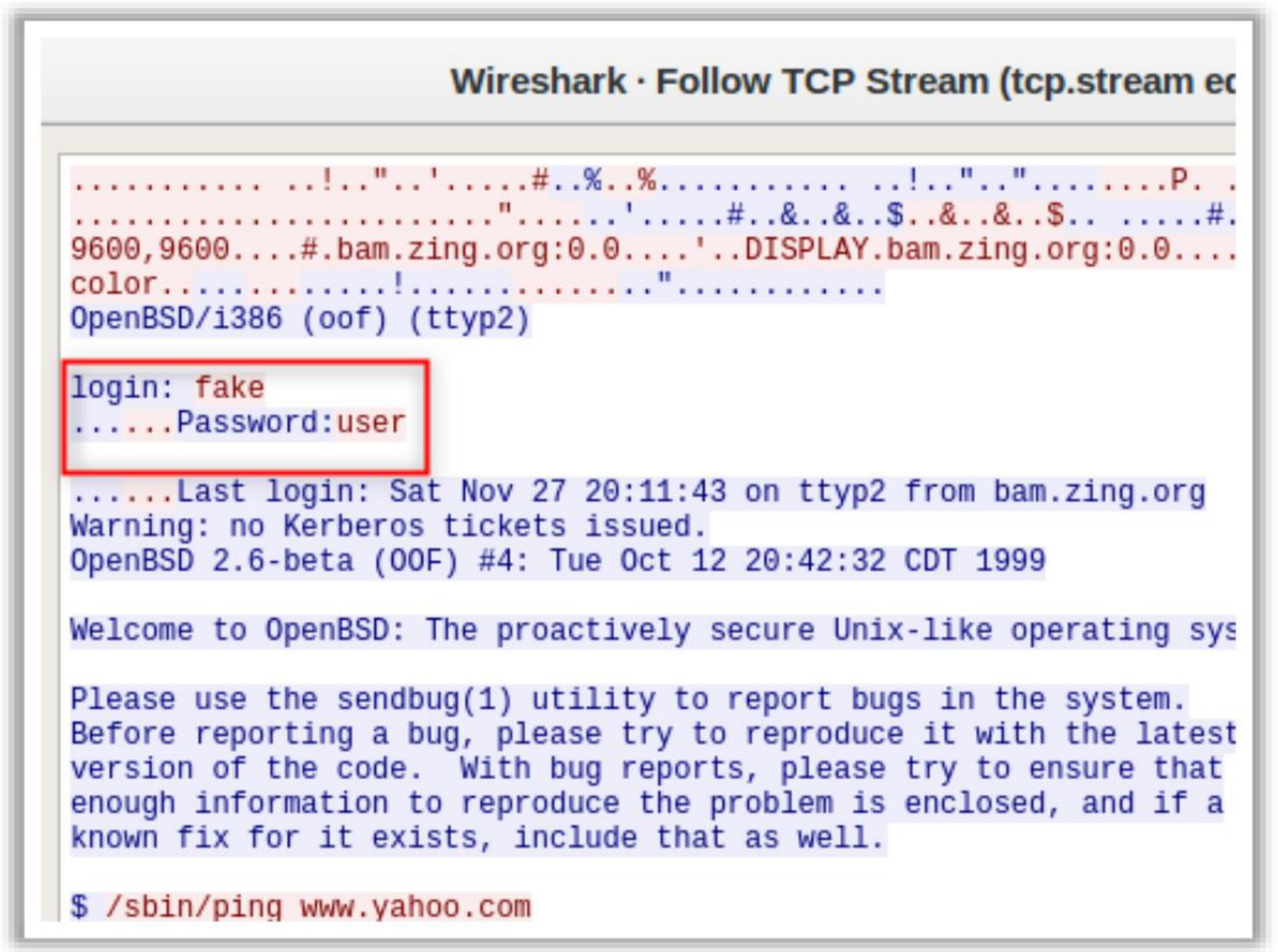
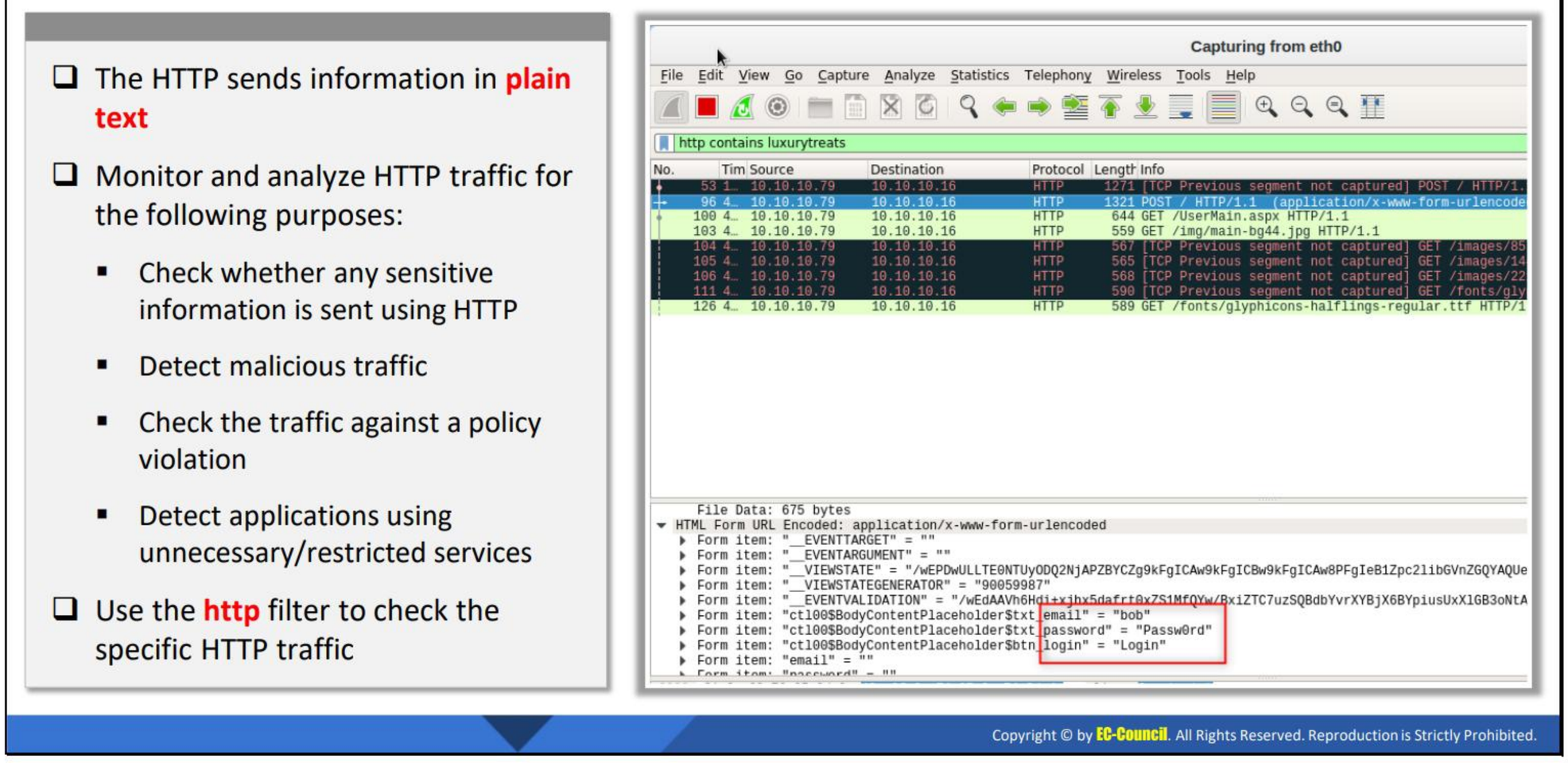


Figure 17.9: Cleartext credential in telnet traffic



Monitoring and Analyzing HTTP Traffic

Applications implementing HTTP send data in cleartext. Implementing HTTP can pose security risks to the organization as sensitive information such as usernames and passwords are sent over as HTTP requests. The attacker can easily sniff the traffic and steal sensitive information for malicious use. Therefore, security professionals must ensure that their HTTP traffic is sent over an encrypted protocol such as HTTP Secure (HTTPS). Simultaneously, they should monitor applications and ensure that they do not send data over HTTP. Monitoring the HTTP traffic also helps detect the volume of HTTP traffic in the network. It also helps detecting malicious traffic, policy violation attempts, applications using unnecessary/restricted services.

Use the http filter to check the specific HTTP traffic.

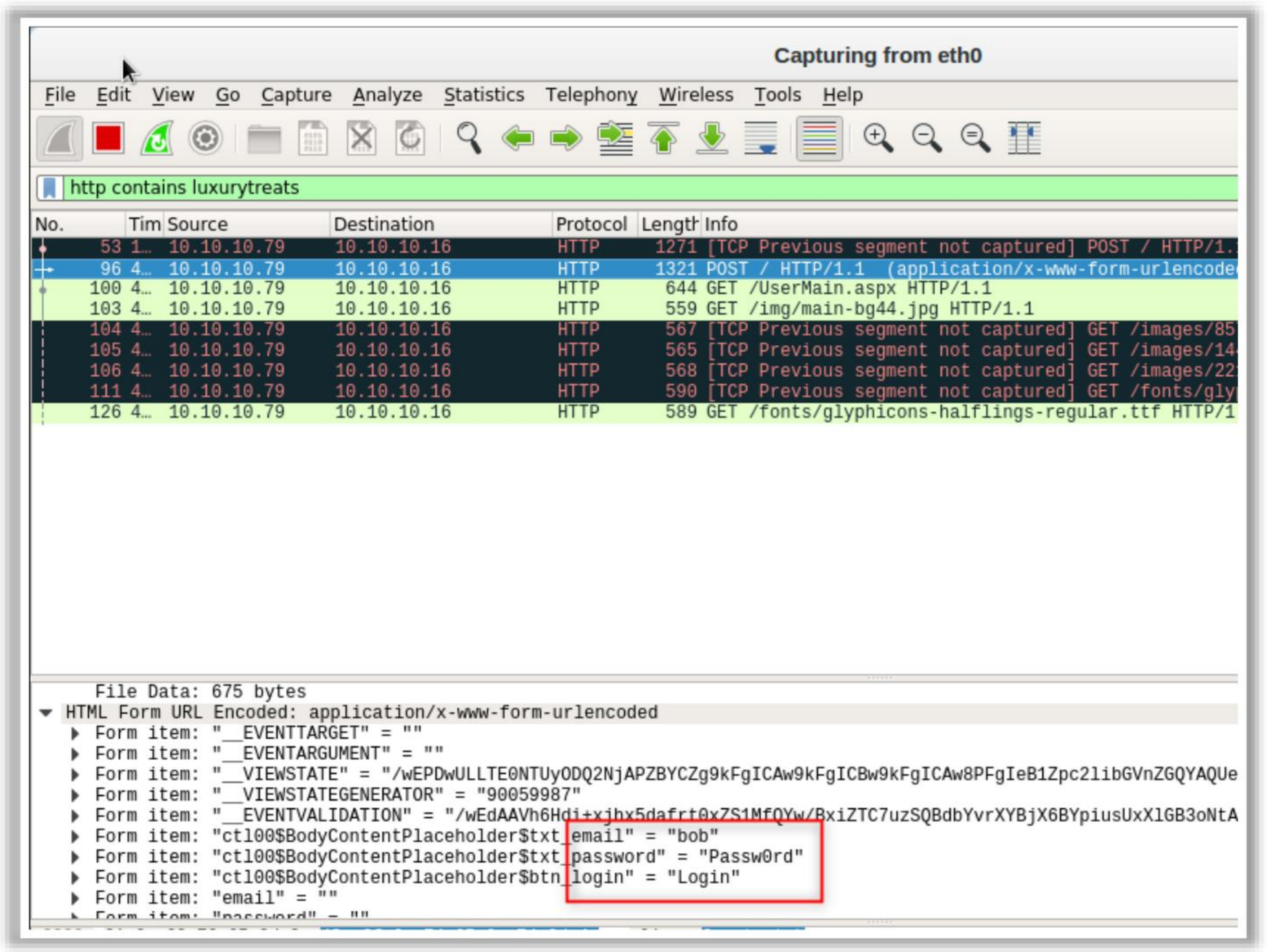
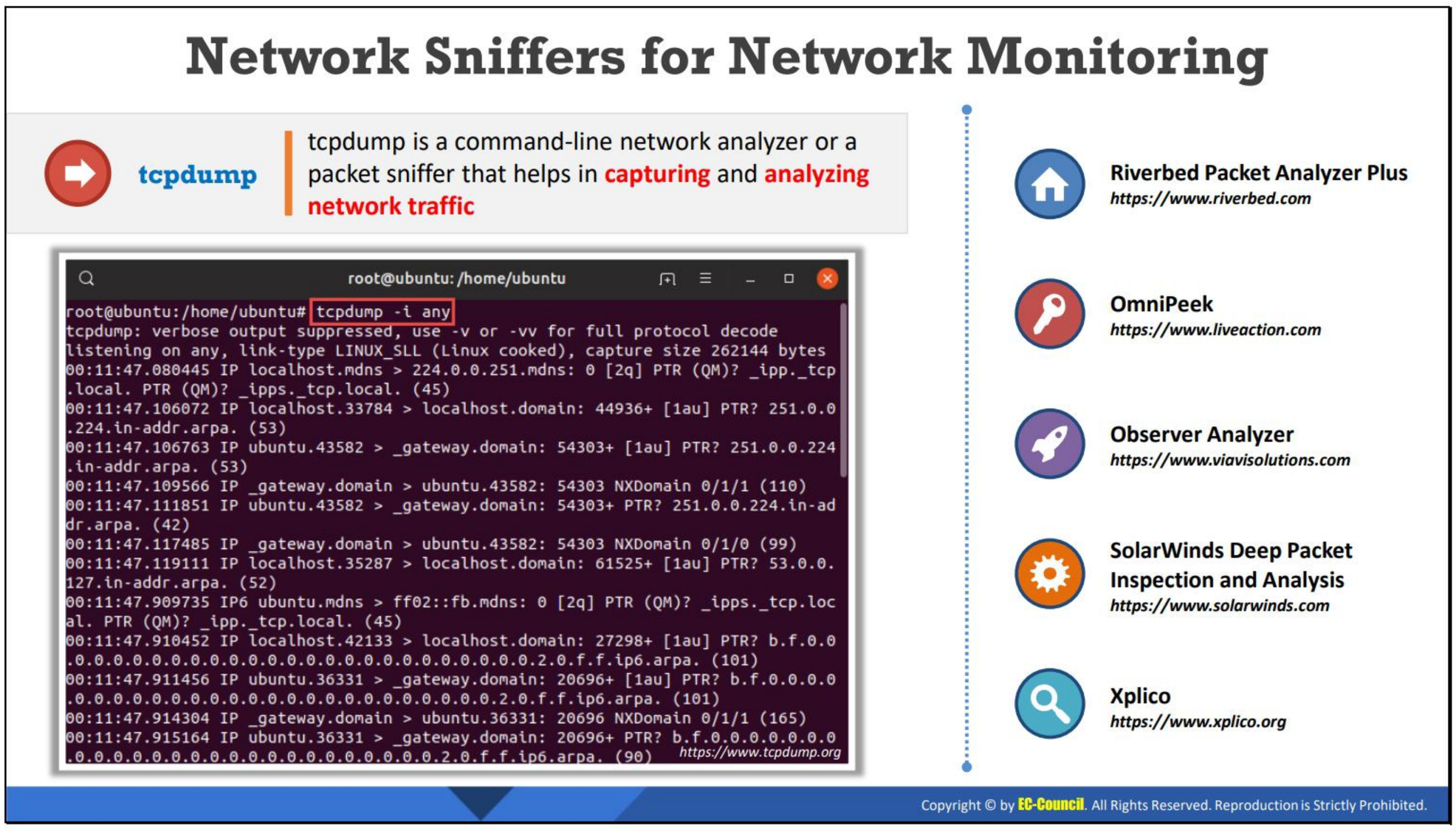


Figure 17.10: Cleartext credential in HTTP traffic



Network Sniffers for Network Monitoring

- **tcpdump**

  Source: *https://www.tcpdump.org*

  tcpdump is a command-line network analyzer or a packet sniffer. Security professionals can use this utility for network monitoring and analysis.

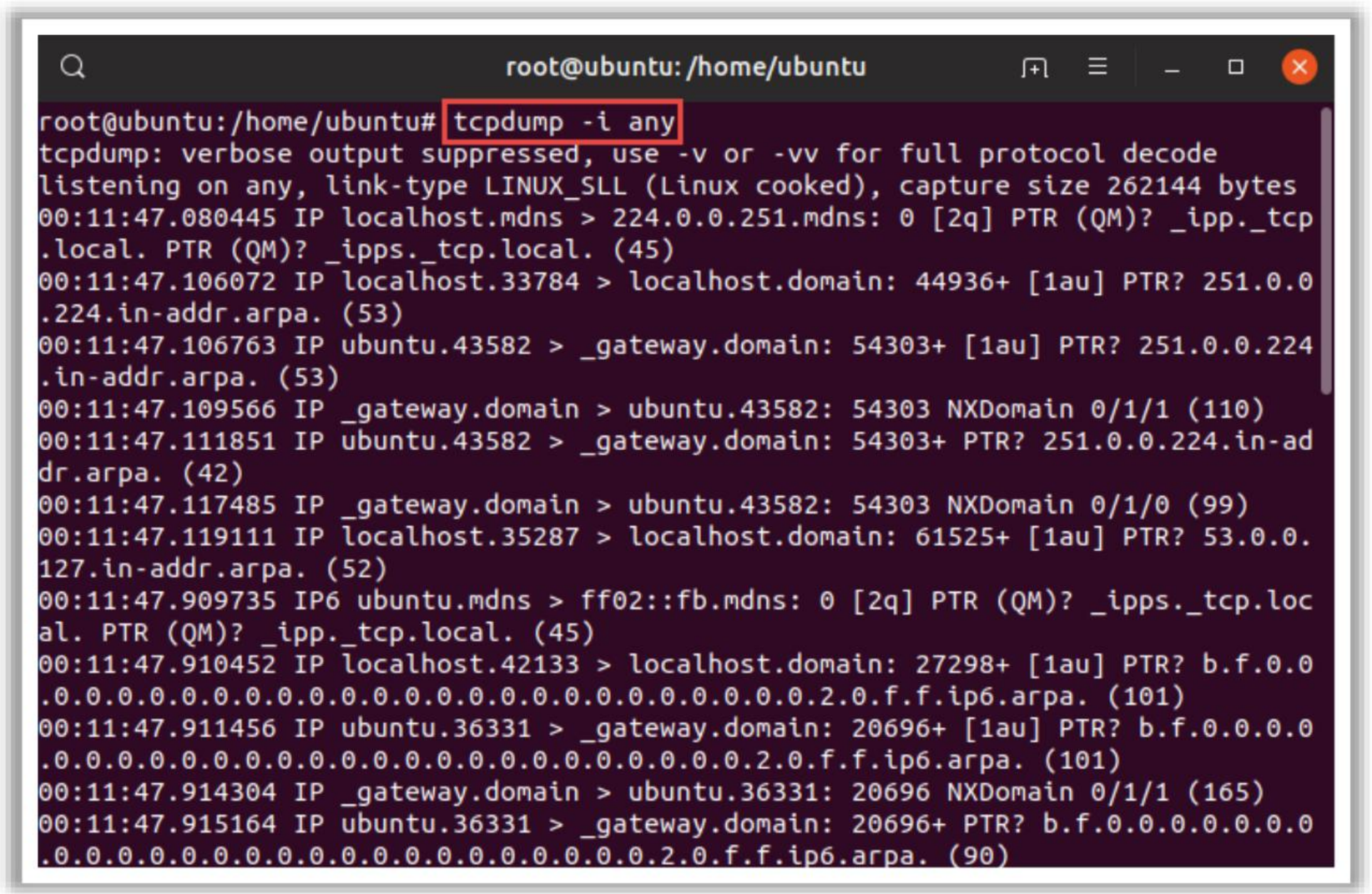
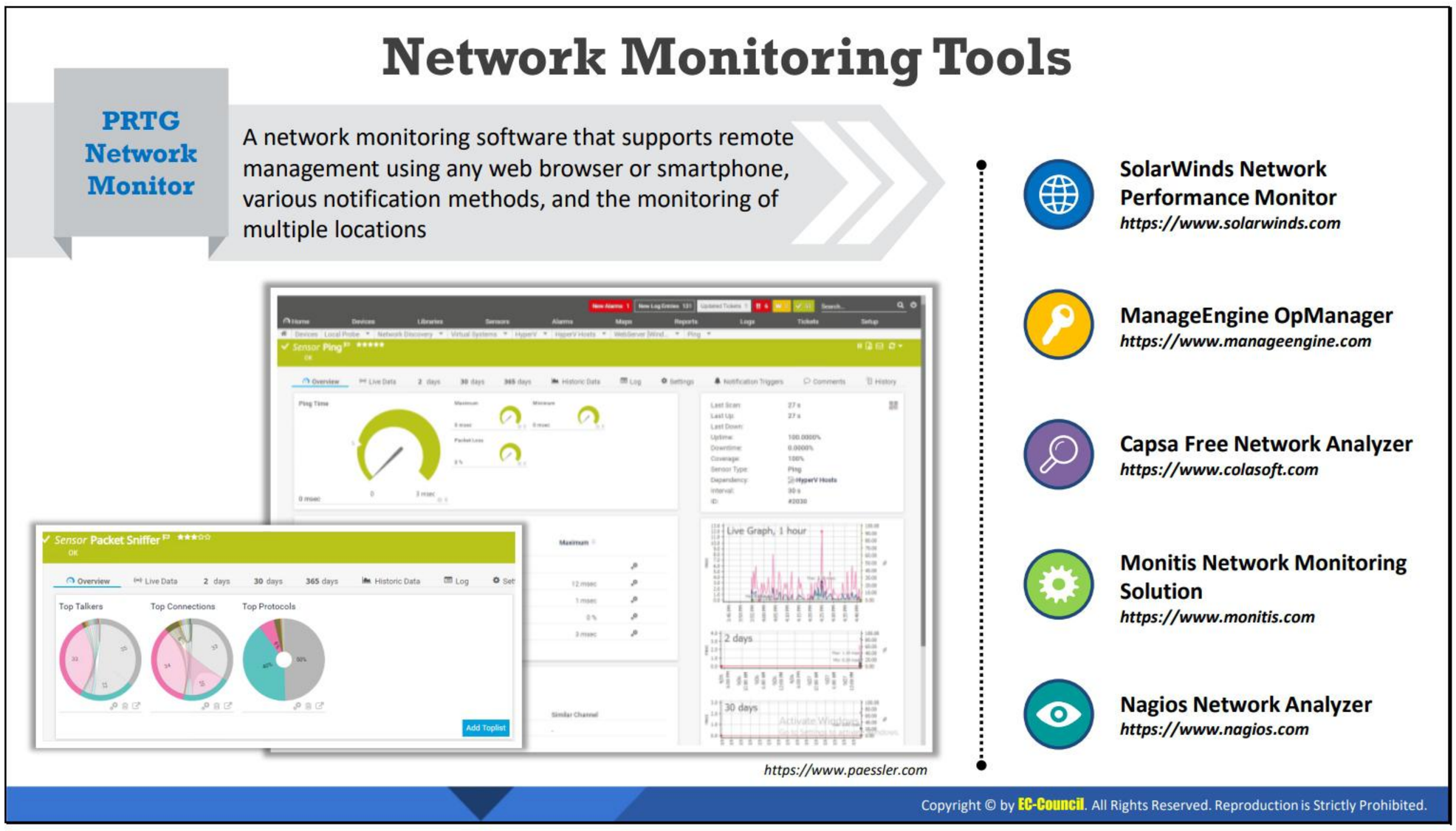


Figure 17.11: Screenshot of tcpdump

Some additional network sniffing tools are as follows:

- Riverbed Packet Analyzer Plus (*https://www.riverbed.com*)
- OmniPeek (*https://www.liveaction.com*)
- Observer Analyzer (*https://www.viavisolutions.com*)
- SolarWinds Deep Packet Inspection and Analysis (*https://www.solarwinds.com*)
- Xplico (*https://www.xplico.org*)



Network Monitoring Tools

- **PRTG Network Monitor**

  Source: *https://www.paessler.com*

  PRTG Network Monitor is a network monitoring software that supports remote management using any web browser or smartphone, various notification methods, and the monitoring of multiple locations. A security professional can use this utility for availability, usage, and activity monitoring, and it covers the entire range from website monitoring to database performance monitoring.

  It helps in the following:

  - Avoid bandwidth and performance bottlenecks.
  - Identify applications or servers using up the available bandwidth.
  - Instantly identify sudden spikes caused by malicious code.
  - Reduce the costs of purchasing additional hardware and bandwidth.

  PRTG can collect data for almost anything of interest on the network. It supports multiple protocols for collecting data:

  - Simple Network Management Protocol (SNMP) and Windows Management Instrumentation (WMI)
  - Packet sniffing
  - NetFlow, IP Flow Information Export (IPFIX), jFlow, and sFlow

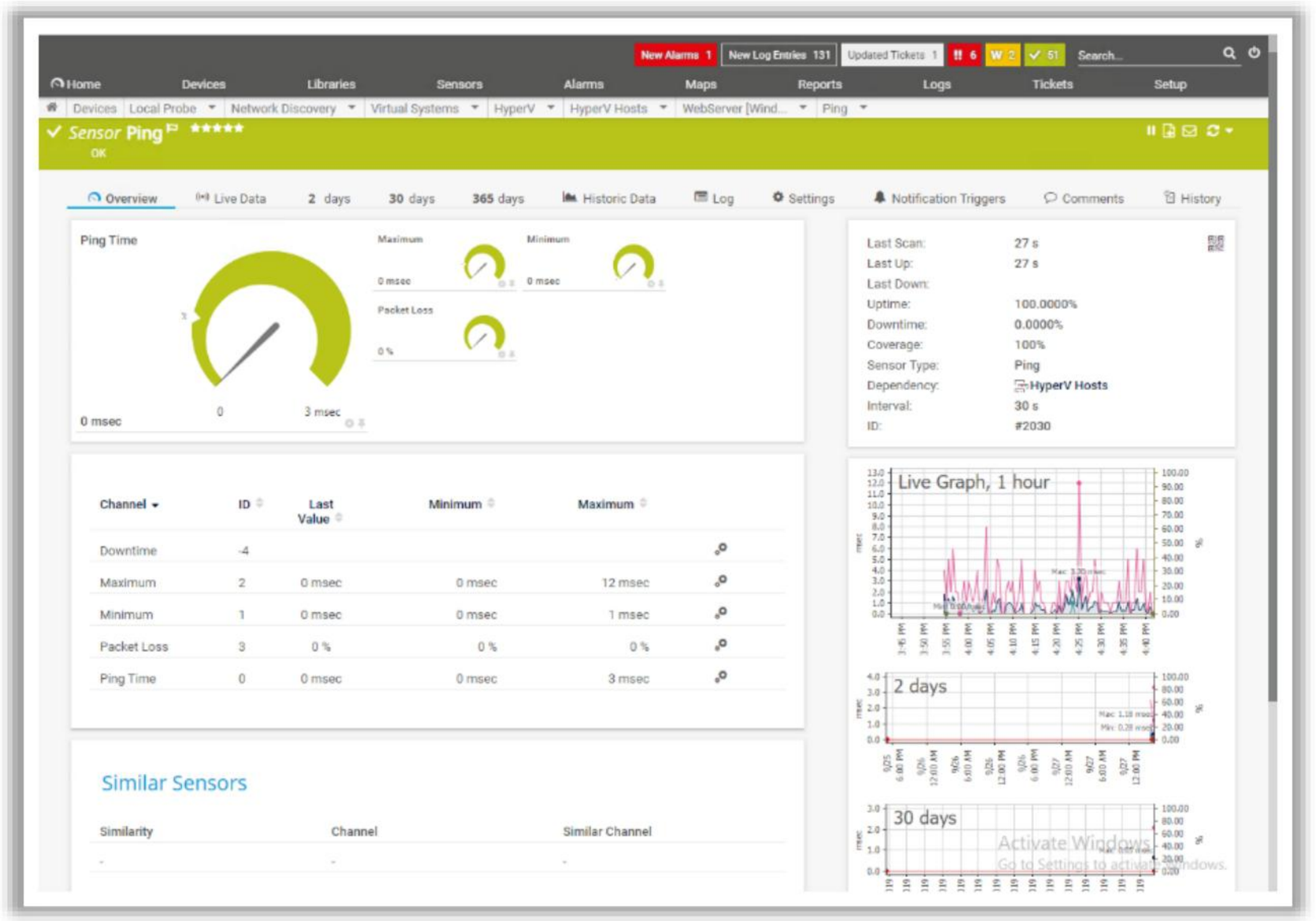


Figure 17.12: Screenshot of PRTG Network Monitor

Figure 17.13: Performance monitoring using PRTG Network Monitor

Some additional network monitoring tools are as follows:

- SolarWinds Network Performance Monitor (*https://www.solarwinds.com*)
- ManageEngine OpManager (*https://www.manageengine.com*)
- Capsa Free Network Analyzer (*https://www.colasoft.com*)
- Monitis Network Monitoring Solution (*https://www.monitis.com*)
- Nagios Network Analyzer (*https://www.nagios.com*)

Module Summary

- This module has discussed the need for and advantages of network traffic monitoring
- It has discussed the network traffic signatures
- It has also discussed the categories of suspicious traffic signatures
- This module also discussed the attack signature analysis techniques and network monitoring for suspicious traffic
- Finally, this module ended with an overview of various network monitoring tools
- In the next module, we will discuss on network logs monitoring and analysis in detail

Module Summary

This module has discussed the need for and advantages of network traffic monitoring. It has discussed the network traffic signatures. It has also discussed the categories of suspicious traffic signatures. This module also discussed the attack signature analysis techniques and network monitoring for suspicious traffic. Finally, this module ended with an overview of various network monitoring tools.

In the next module, we will discuss on network logs monitoring and analysis in detail.