NETWORKING

OSI MODEL

OSI model overview: The OSI (Open Systems Interconnection) model is a conceptual framework by ISO that explains how network communication works using seven layers. It is theoretical but essential for understanding networking and troubleshooting.

Layer 1 Physical: Handles physical data transmission using electrical, optical, or wireless signals such as Ethernet cables, fiber optics, and WiFi bands.

Layer 2 Data Link: Manages communication between devices on the same network segment using MAC addresses. Examples include Ethernet (802.3) and WiFi (802.11).

Layer 3 Network: Responsible for logical addressing and routing between different networks. Protocols include IP, ICMP, and VPN-related protocols.

Layer 4 Transport: Provides end-to-end communication between applications, handling segmentation, flow control, and reliability. Examples are TCP and UDP.

Layer 5 Session: Establishes, maintains, and synchronizes communication sessions between applications. Examples include NFS and RPC.

Layer 6 Presentation: Ensures data is readable by handling encoding, encryption, and compression. Examples include Unicode, MIME, JPEG, and PNG.

Layer 7 Application: Provides network services directly to user applications. Common protocols include HTTP, FTP, DNS, SMTP, POP3, and IMAP.

Why it matters: The OSI model helps clearly understand how data moves through a network and explains real-world concepts such as Layer 3 routing and Layer 7 firewalls.

TCP/IP MODEL

TCP/IP model overview: The TCP/IP model is an implemented networking model developed in the 1970s by the U.S. Department of Defense. It was designed to keep networks operational even if parts fail, making it resilient to disruptions such as attacks. This reliability is achieved through adaptive routing that responds to network topology changes.

-Layer structure (top to bottom): The TCP/IP model groups OSI layers differently and focuses on practical implementation rather than theory.

-Application Layer: Combines OSI layers 5, 6, and 7 (session, presentation, application). It provides services directly to user applications. Examples include HTTP, HTTPS, FTP, SMTP, POP3, IMAP, Telnet, and SSH.

-Transport Layer: Corresponds to OSI layer 4 and enables end-to-end communication between applications. It manages data transfer reliability and flow control using TCP and UDP.

-Internet Layer: Corresponds to OSI layer 3 and handles logical addressing and routing between networks. Protocols include IP, ICMP, and IPSec.

-Link Layer: Corresponds to OSI layer 2 and manages data transfer over the local network using technologies such as Ethernet (802.3) and WiFi (802.11).
Alternative five-layer model: Many modern textbooks extend TCP/IP into five layers by separating the Physical layer. The five layers are Application, Transport, Network, Link, and
Physical.

-Why it matters: The TCP/IP model is the foundation of the Internet and real-world networking. Understanding it helps explain how IP, TCP, and UDP work together to move data reliably across interconnected networks.

IP ADDRESSES AND SUBNETS

-IP address basics: An IP address uniquely identifies a host on a network so other devices can communicate with it. IPv4 addresses like 192.168.0.1 or 172.16.159.243 are the most common and are usually implied when “IP” is mentioned.

-IPv4 structure: An IPv4 address consists of 4 octets (32 bits total). Each octet is 8 bits and ranges from 0–255.This gives roughly 2^{32} possible addresses, though some are reserved.

-Network and broadcast addresses: The first address in a subnet (ending in .0) is the network address. The last address (ending in .255 in a /24 subnet) is the broadcast address. Broadcast traffic reaches all hosts in that subnet.

-Subnet masks and CIDR: A subnet mask like 255.255.255.0 is equivalent to /24./24 means the first 24 bits are fixed as the network portion. Valid host range for 192.168.66.0/24 is 192.168.66.1–192.168.66.254.

-Viewing IP configuration: On Windows use ipconfig. On Linux/Unix use ifconfig or ip a s. These commands show IP address, subnet mask, broadcast address, MAC address, and interface status.

-Public vs private IP addresses: Public IPs are globally reachable on the Internet. Private IPs are used inside local networks and cannot be reached directly from the Internet.

-Private IP ranges (RFC 1918): 10.0.0.0–10.255.255.255 (10/8),172.16.0.0–172.31.255.255 (172.16/12),192.168.0.0–192.168.255.255 (192.168/16).

-NAT concept: Private IP devices access the Internet through a router with a public IP using Network Address Translation (NAT).

-Routing fundamentals: Routers operate at OSI Layer 3. They examine destination IP addresses and forward packets toward the best next network. Packet delivery usually involves multiple routers before reaching the destination.

UDP AND TCP

-Transport layer purpose: IP identifies the host, but transport protocols enable communication between processes on those hosts using port numbers.

-Port numbers: Used to identify sending and receiving processes. Range is 1–65535 (2 octets); port 0 is reserved.

-UDP (User Datagram Protocol): Connectionless Layer 4 protocol. No connection setup, delivery confirmation, or reliability mechanisms. Fast and lightweight. Best suited for speed-critical tasks. Real-world analogy is standard mail with no delivery confirmation.

-UDP characteristics: No acknowledgments, no sequencing, no guarantee of delivery. Lower overhead and faster transmission.

-TCP (Transmission Control Protocol): Connection-oriented Layer 4 protocol. Provides reliable, data-ordered, and error-checked delivery. Requires a connection before data transfer.

-TCP reliability mechanisms: Uses sequence numbers to track data and detect loss or duplication. Uses acknowledgment numbers to confirm received data.

-TCP three-way handshake: Connection establishment uses three packets. Client sends SYN, server replies with SYN-ACK, client responds with ACK.

-TCP vs UDP summary prioritizes speed over reliability.TCP prioritizes reliability, order, and correctness over speed.

ENCAPSULATION

-Encapsulation: Each network layer adds its own header (and sometimes a trailer) to data before passing it down, allowing layers to work independently.

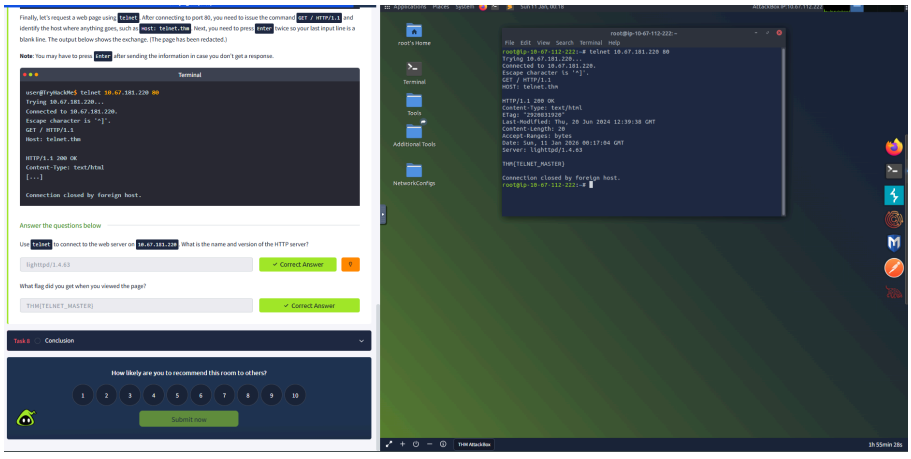
-Process: Application data is wrapped by TCP or UDP into a segment or datagram, then by IP into a packet, and finally by Ethernet or WIFI into a frame.

-Receiving end: The process is reversed as each layer removes its header to recover the original data.

-Life of a packet: User sends a request, the browser creates an HTTP message, TCP establishes a connection and sends data, IP routes packets across routers, and the destination decapsulates the data to reach the application.

TELNET

- **Telnet is a text-based protocol used to connect to services running on TCP ports**
- **It allows manual communication with servers by sending plain text commands**
- **The target VM runs multiple services for testing telnet connections**
- **Echo server runs on port 7 and sends back any text you type**
- **Daytime server runs on port 13 and returns the current date and time before closing**
- **Web server runs on port 80 and serves web pages using HTTP**
- **Echo and daytime services are insecure and enabled only for demonstration**
- **Connecting to port 7 shows basic client-server interaction through echoed input**
- **Connecting to port 13 shows a short-lived TCP connection that closes automatically**
- **Connecting to port 80 allows manual sending of an HTTP GET request**
- **HTTP requests require a GET line, Host header, and a blank line to receive a response**
- **Telnet sessions can be exited using Ctrl+] followed by the quit command**



NETWORKING ESSENTIALS

DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

-Devices need basic network settings to access a network: IP address with subnet mask, gateway, and DNS server

-Manually configuring these settings is common for servers but impractical for mobile devices

-Automatic configuration prevents IP address conflicts and saves time

-This automation is handled by DHCP (Dynamic Host Configuration Protocol)

-DHCP is an application-layer protocol that uses UDP

-The DHCP server listens on UDP port 67 and the client uses UDP port 68

-Most laptops and smartphones use DHCP by default

-DHCP works using four steps called DORA

-Discover: the client broadcasts a request to find a DHCP server

-Offer: the server replies with an available IP address and configuration

-Request: the client accepts the offered IP address

-Acknowledge: the server confirms and assigns the IP to the client

-During Discover and Request, the client has no IP and uses 0.0.0.0 as source

-Broadcast address 255.255.255.255 is used to reach all DHCP servers

-Broadcast MAC address ff: ff: ff: ff: ff: ff is used at the link layer

-The server sends responses directly to the client's MAC address

-After DHCP completes, the device receives

-An IP address lease

-A default gateway for routing traffic

-A DNS server for domain name resolution

ARP (ADDRESS RESOLUTION PROTOCOL)

- **IP packets must be encapsulated inside data link frames to travel over layer 2**
- **Common data link technologies are Ethernet (IEEE 802.3) and WiFi (IEEE 802.11)**
- **To send an Ethernet or WiFi frame, the sender must know the destination MAC address**
- **IP addresses identify hosts at layer 3, while MAC addresses identify devices at layer 2**
- **MAC addresses are 48-bit values written in hexadecimal (example:7C:DF:A1:D3:8C:5C)**
- **Devices normally work only with IP addresses and do not track MAC addresses continuously**
- **Even with DHCP, only IPs of the gateway and DNS are known, not their MACs initially**
- **Communication within the same local network requires MAC address resolution**
- **Ethernet frame header contains destination MAC, source MAC, and type (IPv4, ARP, etc.)**
- **ARP (Address Resolution Protocol) maps an IP address to a MAC address**
- **When a host needs a MAC, it sends an ARP Request as a broadcast**
- **ARP Request destination MAC is ff:ff:ff:ff:ff:ff**
- **The device owning the requested IP replies with an ARP Reply containing its MAC**
- **After ARP resolution, hosts can exchange Ethernet frames directly**
- **ARP messages are not encapsulated in IP or UDP**
- **ARP is carried directly inside Ethernet frames**
- **ARP is often considered layer 2, but it supports layer 3 IP communication**
- **Commands used to observe ARP traffic**
- **tshark -r arp. pcapng -Nn reads packet captures and shows ARP details**
- **tcpdump -r arp. pcapng -n -v displays ARP requests and replies**

INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

- **ICMP is used for network diagnostics and error reporting**
- **ping checks if a host is reachable and measures RTT**
- **Uses ICMP Echo Request (Type 8) and Echo Reply (Type 0)**
- **Displays packet loss and RTT statistics**
- **Firewalls may block ICMP replies**
- **Command: ping 192.168.11.1 -c 4**
- **traceroute finds the path packets take to a destination**
- **Uses TTL to trigger ICMP Time Exceeded (Type 11) messages**
- **Each hop represents a router**
- **means no ICMP reply received**

- Route can change between runs
- Command (Linux): traceroute example.com
- Command (Windows): tracert example.com

ROUTING

- Internet routing is needed to deliver packets between different networks
- Routers decide how to forward packets based on routing algorithms
- Multiple paths (routes) can exist between a source and a destination
- Each router selects the best next link to move the packet closer to its destination
- Routing algorithms determine efficiency, reliability, and reachability
- Routing protocols help routers share network information and build routing tables
- OSPF (Open Shortest Path First)
- Link-state routing protocol
- Routers share full network topology information
- Each router calculates the shortest/best path independently
- Commonly used within large enterprise networks
- EIGRP (Enhanced Interior Gateway Routing Protocol)
- Cisco proprietary protocol
- Hybrid approach combining distance-vector and link-state concepts
- Uses metrics like bandwidth and delay to select best paths
- Fast convergence and efficient routing
- BGP (Border Gateway Protocol)
- Core routing protocol of the Internet
- Used between autonomous systems (ISPs, large networks)
- Exchanges routing information between different organizations
- Focuses on policy-based routing rather than shortest path
- RIP (Routing Information Protocol)
- Simple distance-vector routing protocol
- Uses hop count as the routing metric
- Maximum of 15 hops allowed
- Suitable only for small networks due to limited scalability

NETWORK ADDRESS TRANSLATION

- IPv4 supports about 4 billion addresses, which is insufficient due to rapid growth of Internet-connected devices
- Network Address Translation (NAT) was introduced to mitigate IPv4 address exhaustion

- **NAT allows multiple private devices to share a single (or very few) public IP address**
- **A company or home network can connect many devices to the Internet using one public IP**
- **Private IP addresses are used inside the local network, public IP addresses are used externally**
- **NAT-enabled routers track active connections using a translation table**
- **The translation table maps internal IP address and port to external IP address and port**
- **Example: a device with private IP 192.168.0.129 and source port 15401 appears to the Internet as public IP 212.3.4.5 with port 19273**
- **NAT modifies packet headers transparently so communication works seamlessly**
- **Unlike simple routing, NAT must maintain state to correctly forward return traffic**
- **NAT significantly reduces public IP address usage while enabling Internet connectivity**

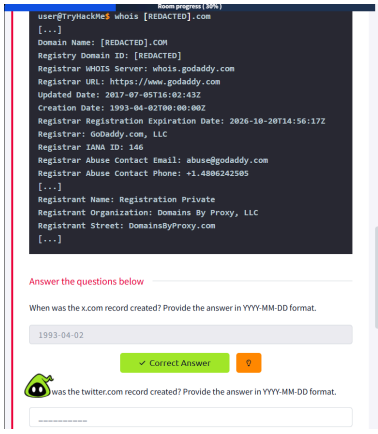
DNS REMEMBERING THE ADDRESS

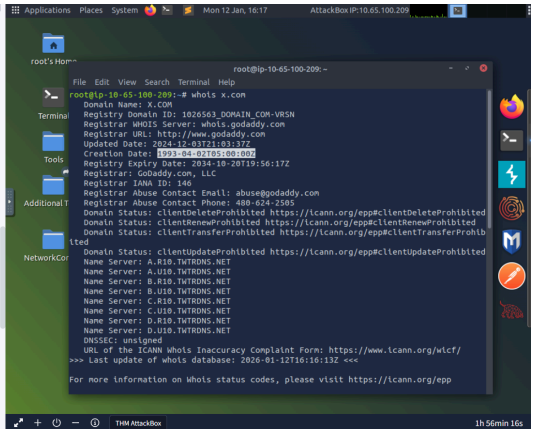
- **DNS maps human-readable domain names to IP addresses so users don't need to memorize IPs**
- **DNS operates at the Application Layer (OSI Layer 7)**
- **Uses UDP port 53 by default and TCP port 53 as fallback**
- **Common DNS record types:**
- **A record maps a hostname to an IPv4 address**
- **AAAA record maps a hostname to an IPv6 address**
- **CNAME record maps one domain name to another domain name**
- **MX record specifies the mail server for a domain**
- **Web browsing typically queries A or AAAA records**
- **Email delivery queries MX records to find the correct mail server**
- **Command-line DNS lookup tool: nslookup**
- **Example command: nslookup www.example.com**
- **DNS queries may request both A and AAAA records**
- **Packet analysis can be done using tshark**
- **Example command: tshark -r dns-query.pcapng -Nn**
- **Correct answers:**
- **IPv6 DNS record type: AAAA**
- **Email server DNS record type: MX**

WHOIS

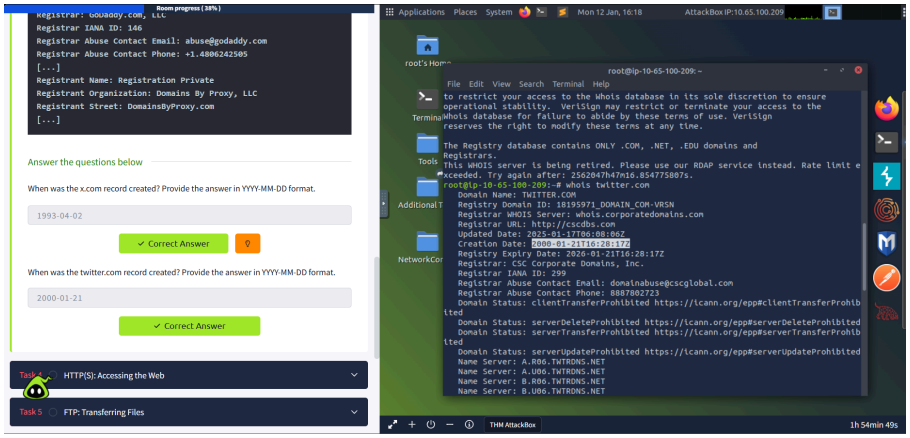
- Domain owners control DNS records such as A, AAAA, MX, and others for their domain
- Registering a domain grants authority to configure its DNS records
- Domains are registered for one or more years with an annual fee
- Accurate registrant contact information is required during registration
- Registrant details are stored in publicly accessible WHOIS records
- WHOIS is pronounced “who is” and is not an acronym
- WHOIS records include domain creation date, update date, and expiration date
- WHOIS records may list registrant name, organization, address, phone, and email
- Privacy protection services can hide personal details in WHOIS records
- Even with privacy enabled, registrar information remains visible
- WHOIS lookups can be done using online tools or the whois command
- Example command: whois example.com
- WHOIS output can reveal the registrar, registrar abuse contacts, and registration dates

Task-3: - (Q)1





Task-3: - (Q)2

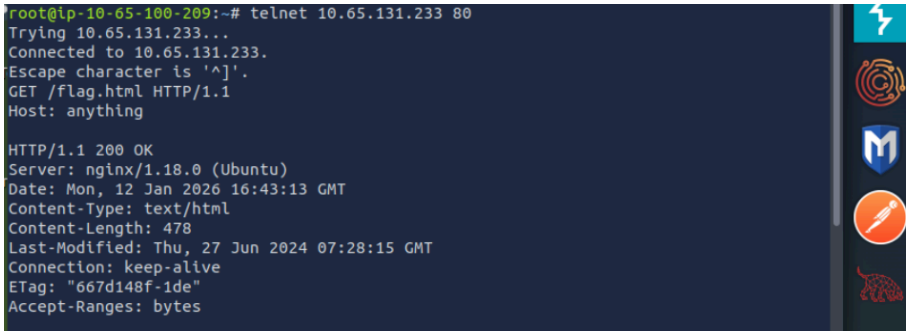


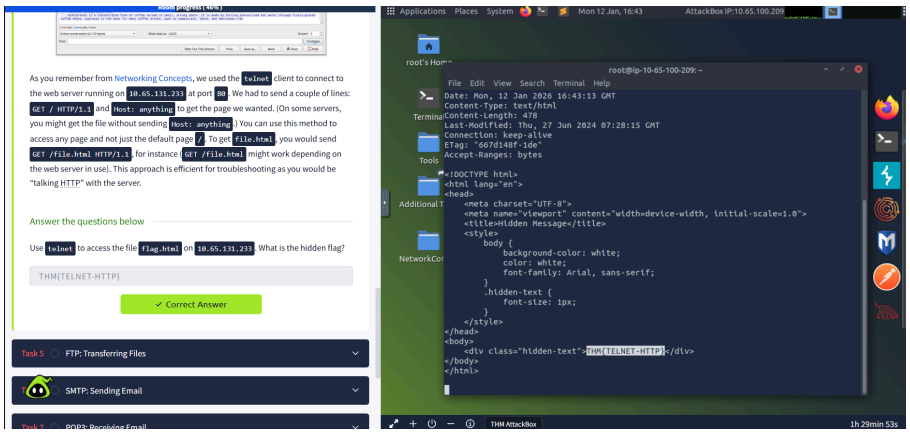
HTTPS (ACCESSING THE WEB)

- Web browsers mainly use HTTP and HTTPS to communicate with web servers
- HTTP stands for Hypertext Transfer Protocol; HTTPS is the secure version
- HTTP/HTTPS rely on TCP for reliable data transfer
- Common HTTP methods include:
- GET retrieves data such as web pages or images
- POST sends data to the server, such as form submissions or file uploads
- PUT creates or updates resources on the server
- DELETE removes a specified resource from the server
- Default ports are TCP 80 for HTTP and TCP 443 for HTTPS
- Alternative ports like 8080 and 8443 may also be used
- Browsers exchange much more data than what users see on the page
- Tools like Wireshark can capture and inspect HTTP requests and responses
- HTTP headers reveal details like server version and last-modified time
- Telnet can be used to manually interact with a web server over HTTP
- Example telnet request includes sending: GET / HTTP/1.1 and Host: header

- Specific files can be requested using paths like GET /file.html HTTP/1.1
- Manually sending HTTP requests is useful for learning and troubleshooting

Task-4: - (Q)1



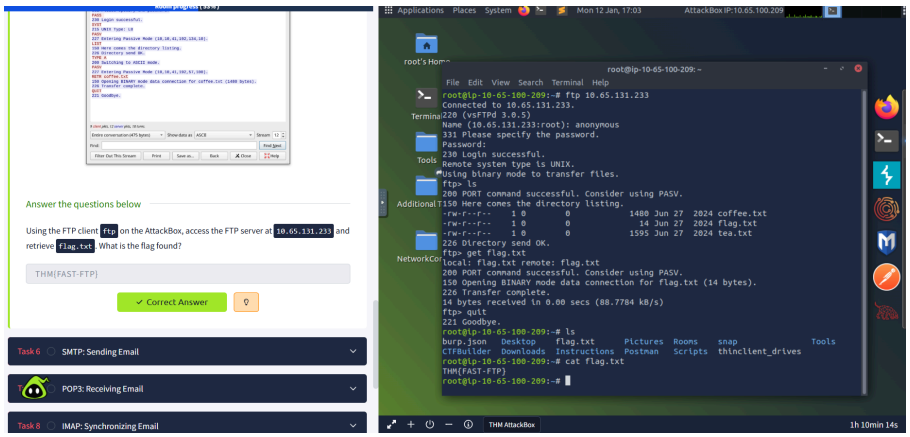


FTP

- FTP is designed specifically for transferring files, unlike HTTP which retrieves web pages
- FTP is generally more efficient and faster for file transfers under equal conditions
- FTP uses a command-and-control connection plus a separate data connection
- Default FTP control connection runs on TCP port 21
- File data is transferred over a separate TCP connection
- Common FTP commands include:
- USER sends the username to the server
- PASS sends the password
- RETR downloads a file from the server to the client
- STOR uploads a file from the client to the server
- FTP clients can connect using the command ftp <server_ip>

- Anonymous login allows access without a password when enabled
- The ls command lists files on the remote server
- type ascii switches transfer mode for text files
- get <filename> downloads a file from the server
- quit closes the FTP session
- FTP client commands may differ from protocol commands
- Example: ls on the client is sent as LIST to the server
- Directory listings and file transfers use separate data connections
- Network tools like Wireshark can capture and analyze FTP traffic

TASK 5: - (Q)1



SMTP (SENDING MAILS)

- SMTP (Simple Mail Transfer Protocol) is used to send emails between clients and mail servers, and between mail servers
- SMTP works at the Application Layer (OSI Layer 7)
- Default SMTP port is TCP 25
- SMTP session flow is similar to sending a package at a post office
- The client introduces itself, provides sender and recipient info, then sends the message
- Common SMTP commands:
- HELO / EHLO → starts the SMTP session
- MAIL FROM → specifies the sender's email address
- RCPT TO → specifies the recipient's email address
- DATA → tells the server the email content is about to be sent
- . (dot on a new line) → marks the end of the email body
- QUIT → terminates the SMTP session

- **Telnet can be used to manually interact with an SMTP server for learning and troubleshooting**
- **SMTP servers respond with numeric status codes indicating success or failure**
- **Wireshark can capture SMTP traffic, showing client requests and server responses**
- **Understanding SMTP makes it easier to learn related email protocols like POP3 and IMAP**

POP3: RECEIVING EMAILS

POP3 Overview (Simple Summary)

- **POP3 (Post Office Protocol v3) is used by email clients to retrieve emails from a mail server.**
- **SMTP is used to send emails, while POP3 is used to download/read emails.**
- **Analogy:**
  - **SMTP → Dropping a letter into a post office box 📮**
  - **POP3 → Checking your personal mailbox 📬**
- **POP3 typically runs on TCP port 110.**
- **POP3 requires authentication (username + password).**
- **When POP3 is used over plain text (e.g., telnet):**
  - **Credentials and emails are not encrypted**
  - **Anyone capturing packets (e.g., with Wireshark) can read usernames, passwords, and email contents**
- **This is why secure versions (POP3S over SSL/TLS) are preferred in real environments.**

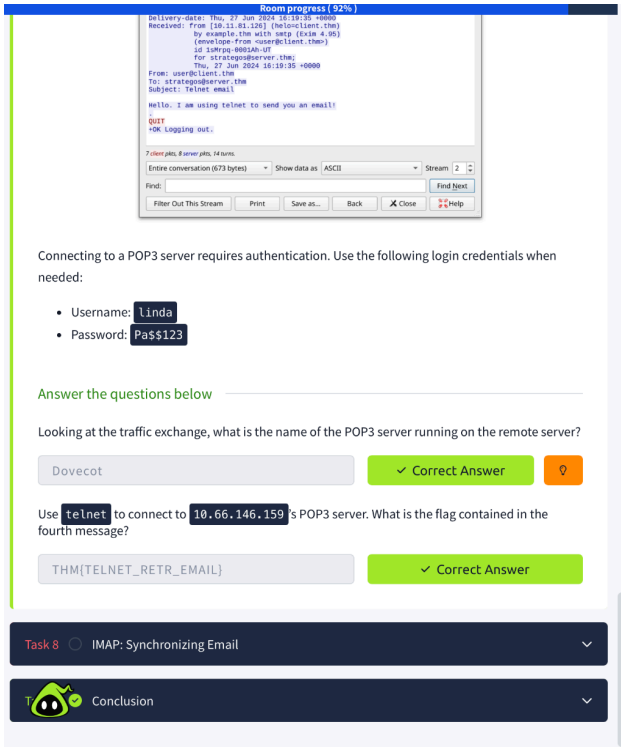
Common POP3 Commands (Bullet Points)

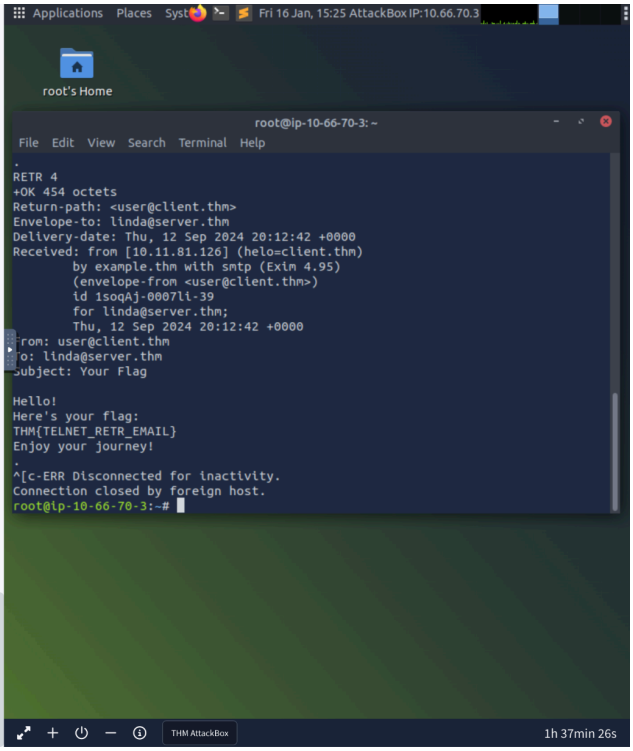
- `USER <username>`
  - **Identifies the user (e.g., `USER linda`)**
- `PASS <password>`
  - **Sends the user's password (e.g., `PASS Pa$$123`)**
- `STAT`
  - **Shows total number of messages and combined size**
- `LIST`
  - **Lists all messages with their individual sizes**
- `RETR <message_number>`
  - **Retrieves (downloads) a specific email**

- **`DELE <message_number>`**
    - **Marks an email for deletion**
- **`QUIT`**
    - **Ends the session and applies changes (like deletions)**

Key Security Takeaway (Important)

- **POP3 over telnet is insecure**
- **Credentials and emails are sent in plain text**
- **Packet sniffers can easily capture:**
    - **Username**
    - **Password**
    - **Email content**
- **Always use encrypted alternatives (POP3S) in real-world systems**





IMAP: SYNCHRONIZING MAILS

IMAP Overview (Simple Summary)

- IMAP (Internet Message Access Protocol) is designed for users who access email from multiple devices(desktop, laptop, phone).
- Unlike POP3, IMAP keeps emails on the server and synchronizes:
  - Read/unread status
  - Deleted messages
  - Moved messages (folders)
- This ensures all devices see the same mailbox state.
- IMAP typically runs on TCP port 143.
- Because messages stay on the server, IMAP uses more server storage than POP3.
- When IMAP is used over telnet (plain text):
  - Usernames, passwords, and emails are visible to packet sniffers
  - Wireshark can clearly capture client commands and server responses
- Secure versions (IMAPS over TLS/SSL) are used in real-world environments.

Common IMAP Commands (Bullet Points)

- `LOGIN <username> <password>`
  - Authenticates the user
- `SELECT <mailbox>`
  - Selects a mailbox/folder (e.g., `inbox`)
- `FETCH <mail_number> <data_item>`
  - Retrieves specific parts of an email
  - Example: `FETCH 3 body[]` → fetches full email #3
- `MOVE <sequence_set> <mailbox>`
  - Moves messages to another folder
- `COPY <sequence_set> <mailbox>`
  - Copies messages to another folder
- `LOGOUT`
  - Ends the IMAP session

POP3 vs IMAP (Key Difference)

- POP3
  - Downloads emails

    - **Often deletes them from the server**
    - **Best for single-device use**
- **IMAP**
    - **Syncs emails across devices**
    - **Keeps emails on the server**
    - **Best for multi-device access**

Security Takeaway

- **IMAP over telnet is not secure**
- **Anyone capturing traffic can read:**
    - **Login credentials**
    - **Email contents**
- **Always prefer IMAPS (encrypted) in production environments**

NETWORK SECURE PROTOCOL

Why Core Networking Protocols Are Insecure

- Protocols like **HTTP, POP3, SMTP, IMAP, and TELNET** work correctly but **do not provide security by default**
- They **do not protect**:
    - **Confidentiality** → attackers can read sensitive data (passwords, credit cards, emails)
    - **Integrity** → attackers can modify data in transit (e.g., changing payment amounts)
    - **Authenticity** → attackers can impersonate legitimate servers (man-in-the-middle attacks)
- Packet sniffers can easily capture and analyze traffic sent using these protocols

---

What Confidentiality, Integrity, and Authenticity Mean

- **Confidentiality**
    - Ensures only intended parties can read the data
    - Without it, attackers can read emails, passwords, and private files
- **Integrity**
    - Ensures data is not altered during transmission
    - Without it, attackers can modify messages or transactions

- **Authenticity**
    - Ensures you are communicating with the legitimate server
    - Without it, fake servers can steal credentials or data

How TLS Fixes the Problem

- **TLS (Transport Layer Security)** is added on top of existing protocols
- TLS provides:
    - Encryption (confidentiality)
    - Message integrity checks
    - Server authentication (certificates)
- When TLS is used, protocols become:
    - **HTTP → HTTPS**
    - **POP3 → POP3S**
    - **SMTP → SMTPS**
    - **IMAP → IMAPS**
- The **“S” stands for Secure**

Secure Remote Access

- **TELNET** is insecure because all data is sent in plain text
- **SSH (Secure Shell)** replaces TELNET
- SSH provides:
    - Encrypted communication
    - Server authentication
    - Secure remote login
- SSH is also **extensible**, allowing secure tunneling and protection of other protocols

---

Key Takeaway

- Core protocols alone are **not safe for sensitive communication**
- **TLS and SSH are essential** for secure online transactions, email, and remote access
- Modern networks rely on **encrypted versions only** to prevent eavesdropping and attacks

TLS

- Past insecurity
  - Earlier, attackers could capture network traffic using packet-capturing tools (promiscuous mode).
  - Chats, emails, and passwords were often sent in cleartext.
  - Users had no way to protect their credentials from being intercepted.
- Need for secure communication
  - This insecurity made private online activities unsafe.
  - Secure communication became essential for trust on the internet.
- Development of SSL/TLS
  - Netscape introduced SSL; SSL 2.0 released in 1995.
  - TLS was developed by the IETF in 1999 as an improvement over SSL.
  - TLS 1.3 (2018) significantly improved security.
  - TLS evolved over decades through lessons learned and improvements.
- Purpose of TLS
  - A cryptographic protocol at the transport layer of the OSI model.
  - Ensures confidentiality (no eavesdropping) and integrity (no tampering).
  - Makes online shopping, banking, email, and messaging possible and safe.
- Protocols secured by TLS
  - HTTP → HTTPS
  - DNS → DoT (DNS over TLS)
  - MQTT → MQTTS
  - SIP → SIPS
  - The “S” indicates security via SSL/TLS.
- TLS certificates
  - Servers (and sometimes clients) use TLS certificates to prove identity.
  - Certificates are issued by Certificate Authorities (CAs) after verification.
  - Systems trust certificates by having CA certificates preinstalled.
  - Most certificates require payment, but Let’s Encrypt provides free ones.
- Self-signed certificates
  - Created without a CA.
  - Do not prove authenticity because no trusted third party verifies them.

HTTPS

- HTTP basics
  - HTTP uses TCP and runs on port 80 by default.
  - All HTTP traffic is sent in cleartext.
  - Anyone capturing packets can read requests, responses, and sensitive data.
- HTTP communication steps
  - Resolve domain name to an IP address.
  - Establish a TCP three-way handshake.

    - Exchange data using HTTP (e.g., `GET / HTTP/1.1`).
    - Close the TCP connection after data transfer.
- Risk of HTTP
    - Packet-capture tools (e.g., Wireshark) can reconstruct full HTTP sessions.
    - Attackers can easily read and analyze transmitted data.
- HTTPS (HTTP over TLS)
    - HTTPS = HTTP + TLS.
    - Provides encryption and protection against eavesdropping.
- HTTPS communication steps
    - Resolve domain name to an IP address.
    - Establish a TCP three-way handshake.
    - Establish a TLS session (encryption negotiation).
    - Exchange HTTP data securely over TLS.
- Encrypted traffic visibility
    - After TLS is established, data appears as “Application Data”.
    - Traffic on port 443 is encrypted and unreadable without keys.
    - Packet contents appear as gibberish when captured.
- Encryption keys
    - Without the TLS encryption key, packet contents cannot be read.
    - Providing the decryption key to Wireshark allows viewing normal HTTP traffic again.
- Key takeaway
    - TLS secures HTTP without changing TCP or IP.
    - HTTP behaves the same way, but its data is encrypted inside TLS.
    - This allows secure communication over an otherwise insecure network.

SMTPS, POP3S AND IMAP’S

- TLS for email protocols
    - Adding TLS to SMTP, POP3, and IMAP works the same way as adding TLS to HTTP.
    - Appending “S” means the protocol is secured with TLS:
        - SMTP → SMTPS
        - POP3 → POP3S
        - IMAP → IMAPS
- Similarity to HTTPS
    - These secure protocols function like HTTP over TLS.
    - Data is encrypted, protecting confidentiality and integrity.
    - Same security benefits and reasoning as HTTPS.
- Default ports (insecure versions)
    - HTTP: 80

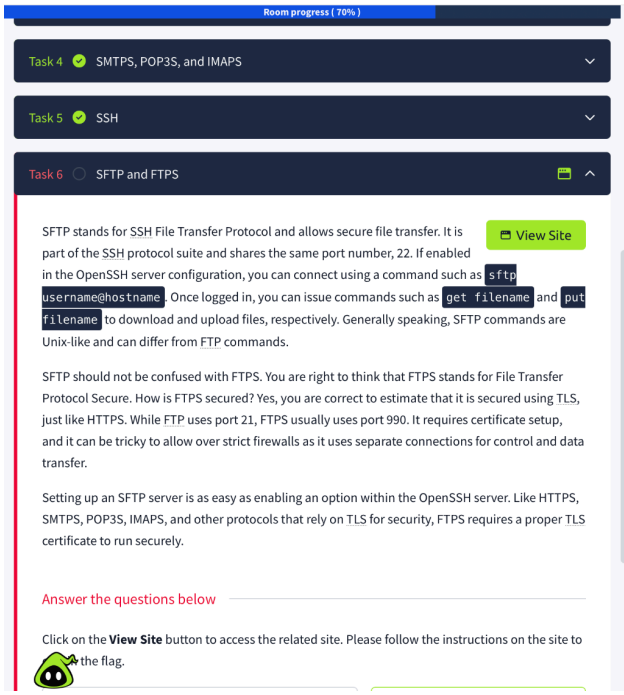
    - SMTP: 25
    - POP3: 110
    - IMAP: 143
- Default ports (secure versions over TLS)
    - HTTPS: 443
    - SMTPS: 465 and 587
    - POP3S: 995
    - IMAPS: 993
- Broader use of TLS
    - TLS can be applied to many other protocols.
    - The advantages remain the same: secure communication over insecure networks.

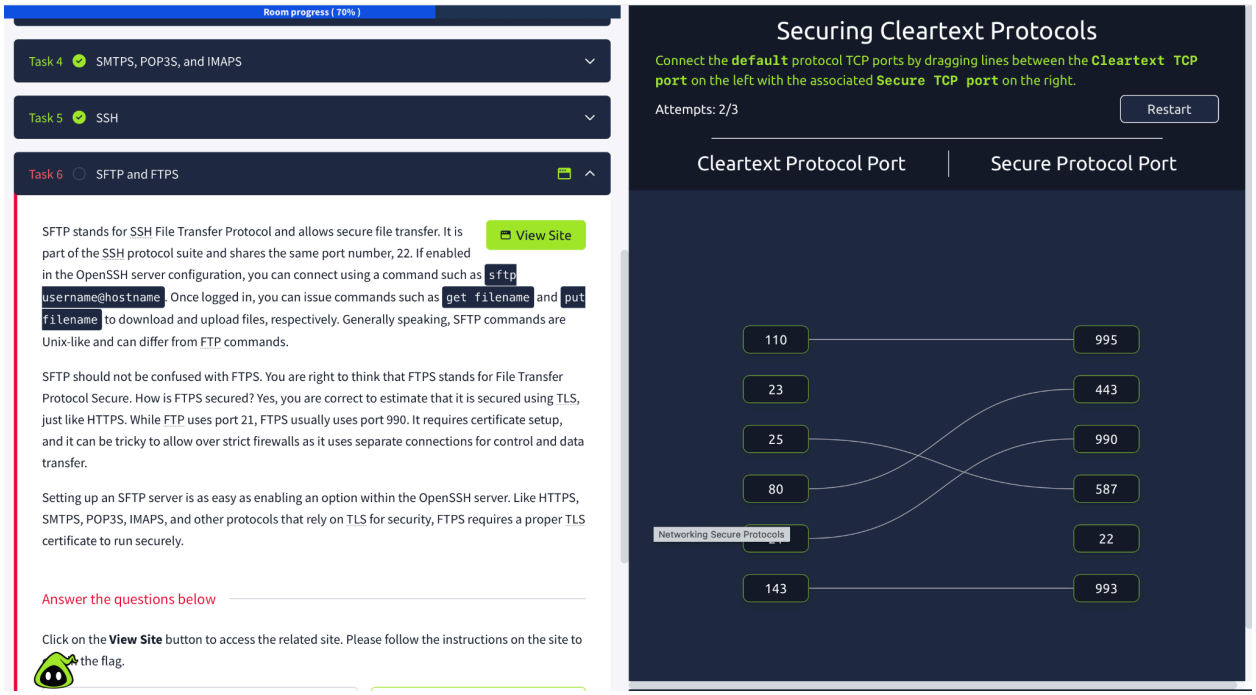
SSH

- Problem with Telnet
    - TELNET sends all traffic, including usernames and passwords, in cleartext.
    - Anyone monitoring the network can easily capture login credentials.
    - This made TELNET insecure for remote administration.
- Introduction of SSH
    - SSH (Secure Shell) was created by Tatu Ylönen in 1995.
    - SSH-1 released as freeware in 1995.
    - SSH-2, a more secure version, defined in 1996.
    - OpenSSH (open-source SSH implementation) released in 1999 by OpenBSD.
    - Most modern SSH clients are based on OpenSSH.
- Key benefits of OpenSSH
    - Secure authentication
        - Supports passwords, public-key authentication, and two-factor authentication.
    - Confidentiality
        - Provides end-to-end encryption.
        - Warns about new or changed server keys to prevent man-in-the-middle attacks.
    - Integrity
        - Cryptographic checks ensure data is not altered in transit.
    - Tunneling
        - Can securely tunnel other protocols through SSH (VPN-like behavior).
    - X11 forwarding
        - Allows running graphical applications from a remote Unix-like system.
- Using SSH

    - Command format: `ssh username@hostname`
    - If the username matches the local user: `ssh hostname`
    - Password may be requested unless public-key authentication is configured.
- Practical example
    - SSH can be used to run graphical applications remotely (e.g., Wireshark).
    - The `-X` option enables graphical forwarding.
- Default ports
    - TELNET: 23
    - SSH: 22

SFTP AND FTPS





- **SFTP overview**
  - **SFTP = SSH File Transfer Protocol.**
  - **Provides secure file transfer over SSH.**
  - **Part of the SSH protocol suite.**
  - **Uses port 22 (same as SSH).**
- **Using SFTP**
  - **Connect with: `sftp username@hostname`.**
  - **Common commands:**
    - **`get filename` → download a file.**
    - **`put filename` → upload a file.**

    - **Commands are Unix-like and differ from traditional FTP commands.**
- **SFTP vs FTPS**
    - **SFTP is secured using SSH.**
    - **FTPS (File Transfer Protocol Secure) is secured using TLS.**
- **FTPS details**
    - **FTP uses port 21.**
    - **FTPS usually uses port 990.**
    - **Requires TLS certificate configuration.**
    - **Can be difficult with strict firewalls due to separate control and data connections.**
- **Ease of setup**
    - **SFTP: enabled by configuring the OpenSSH server.**
    - **FTPS: requires proper TLS certificates, similar to HTTPS and other TLS-based protocols.**

VPN

- **Purpose of a VPN**
    - **Allows a company to connect offices in different locations.**
    - **Makes remote devices behave as if they are on the main branch network.**
    - **Uses the public Internet instead of expensive private links.**

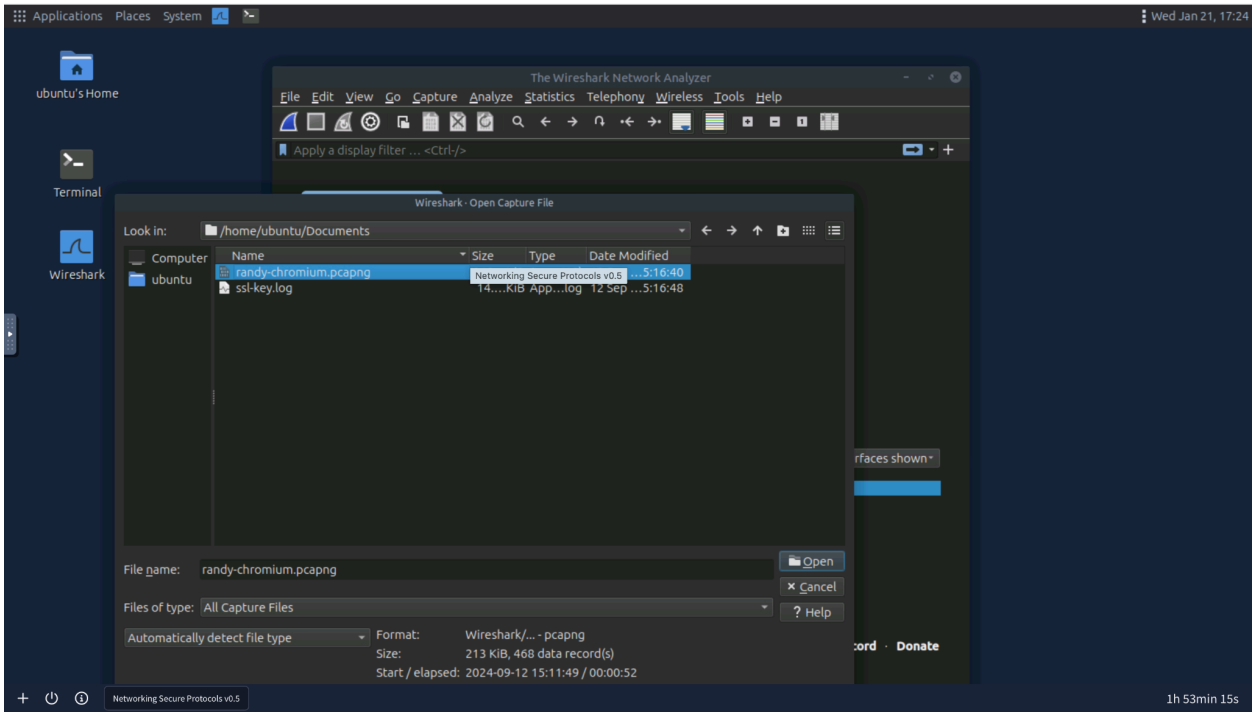
- **Why VPNs are needed**
    - **TCP/IP ensures packet delivery but not confidentiality or integrity.**
    - **Data sent over the Internet can be intercepted or altered.**
    - **VPNs solve this by encrypting traffic.**
- **Key ideas behind VPN**
    - **Virtual: uses the Internet as the transport medium.**
    - **Private: encrypts data to protect it from disclosure and tampering.**
- **Basic VPN requirements**
    - **Internet connectivity.**
    - **A VPN server (usually at the main branch).**
    - **VPN clients (at remote offices or user devices).**
- **Site-to-site VPN**
    - **Remote branches connect to the main branch via VPN tunnels.**
    - **Traffic inside the tunnel is encrypted.**
    - **Decrypted traffic is only visible within the private networks.**
- **Remote-access VPN**
    - **Individual users connect their devices to the main branch.**
    - **Common for employees working from home or travelling.**
- **Traffic routing through VPN**
    - **Often, all Internet traffic is routed through the VPN tunnel.**
    - **External services see the VPN server's IP address, not the user's.**
    - **Local ISPs only see encrypted traffic.**

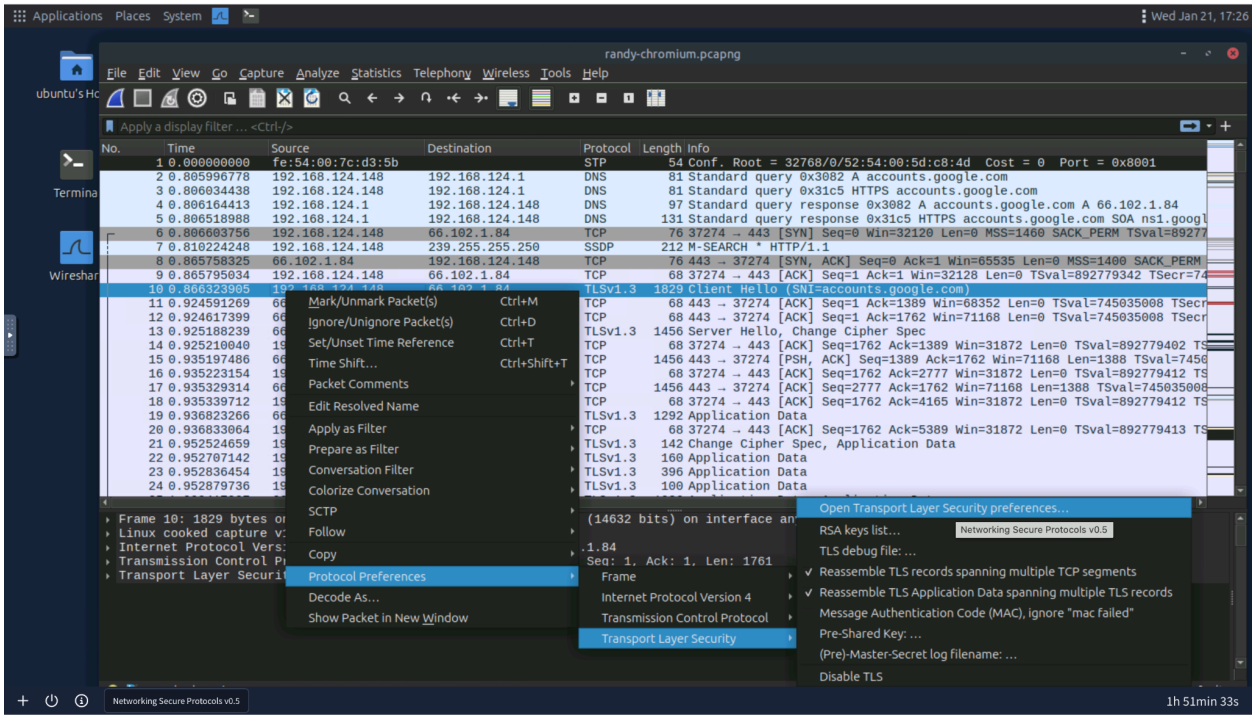
- **Geographical effects**
    - **User appears to be located where the VPN server is.**
    - **Services may change language, content, or access based on that location.**
- **Limitations and caveats**
    - **Some VPNs only provide access to private networks, not full traffic routing.**
    - **Poorly configured VPNs may leak real IP addresses (DNS leaks).**
    - **Additional testing may be required depending on VPN use.**
- **Legal considerations**
    - **VPN use is restricted or illegal in some countries.**
    - **Local laws should always be checked before using a VPN.**

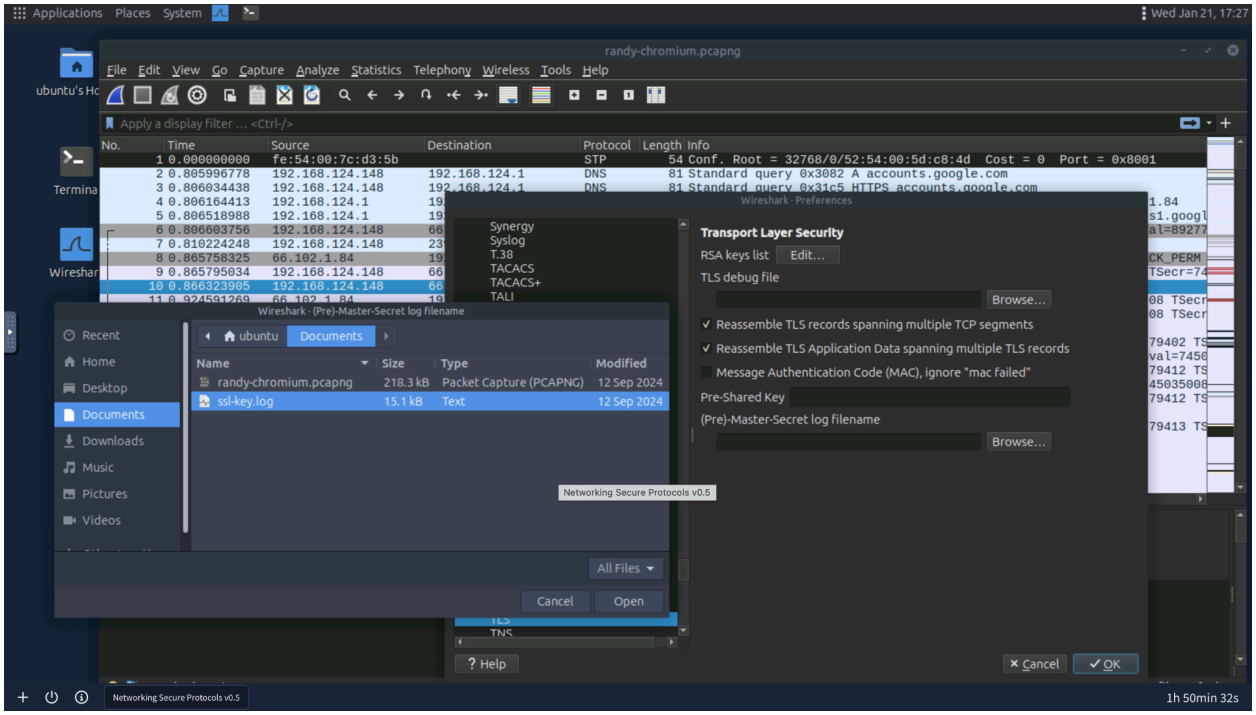
CLOSING

- **Split-screen view**
    - **The machine starts in Split-Screen view.**
    - **If not visible, use the blue “Show Split View” button at the top.**
- **Logging TLS keys**
    - **The browser is configured to log TLS session keys.**
    - **This is done by launching Chromium with:**
        - `chromium --ssl-key-log-file=~/ssl-key.log`
    - **TLS keys are saved to ssl-key.log.**

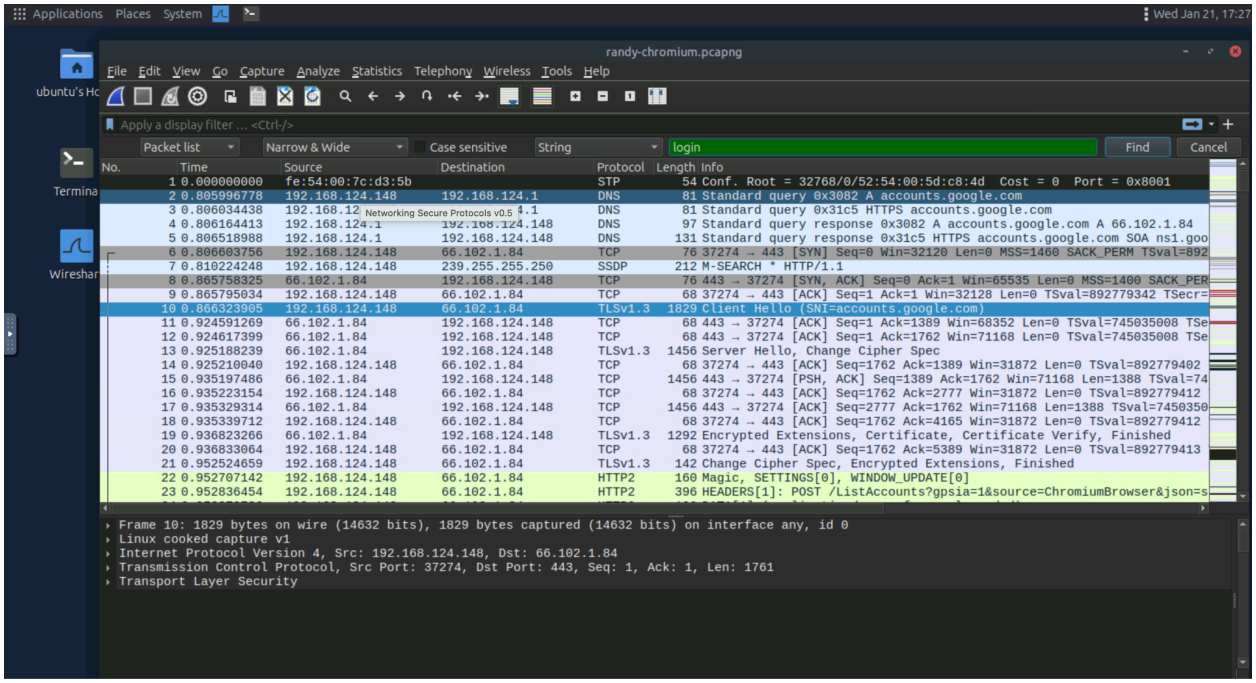
- **Packet capture file**
    - **Capture file name: randy-chromium.pcapng**
    - **Location: Documents folder**
- **Purpose**
    - **Using the TLS keys allows Wireshark to decrypt TLS traffic.**
    - **Decrypted packets can reveal application data, including login credentials.**
- **Configuring Wireshark (steps)**
    - **Open randy-chromium.pcapng in Wireshark.**
    - **Right-click any TLS packet.**
    - **Select Protocol Preferences.**
    - **Choose Transport Layer Security.**
    - **Click Open Transport Layer Security preferences.**
- **Adding the key log file**
    - **In the TLS preferences window, click Browse.**
    - **Select ssl-key.log from the Documents directory.**
    - **Click OK.**
- **Result**
    - **Wireshark decrypts all TLS traffic.**
    - **Encrypted application data becomes readable.**

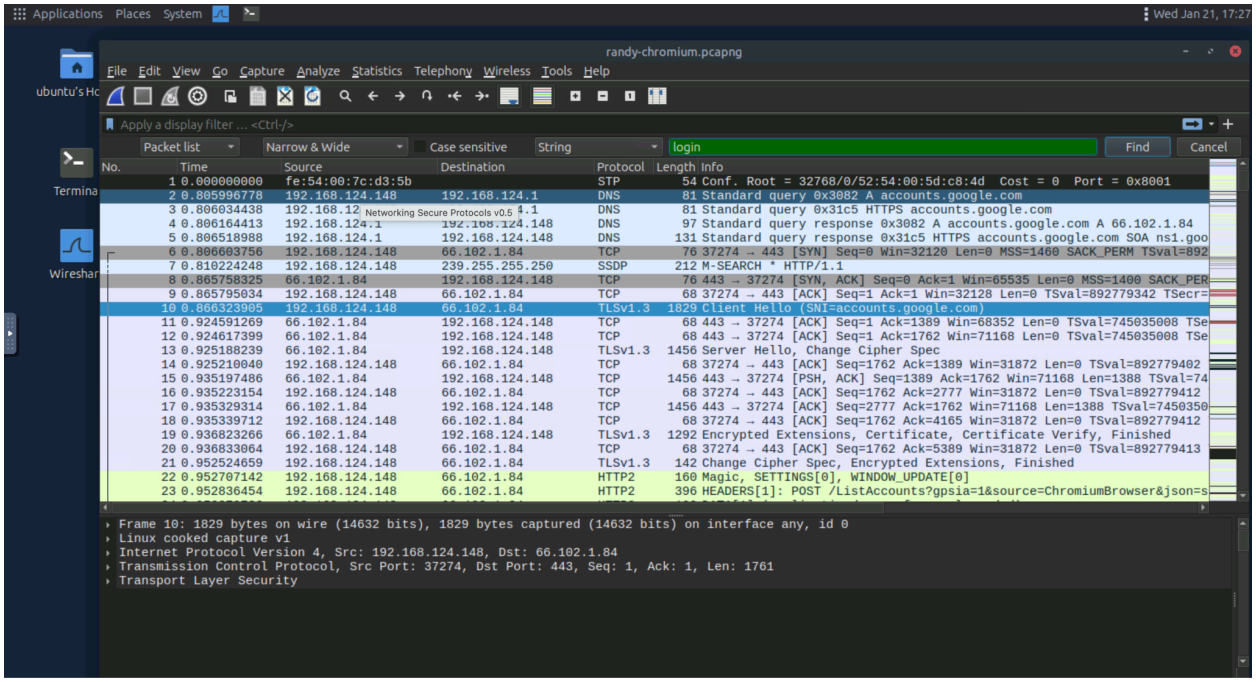

Applications Places System
Wed Jan 21, 17:24
ubuntu's Home
Terminal
Wireshark
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Wireshark · Open Capture File
Look in: /home/ubuntu/Documents
Computer
ubuntu
Name Size Type Date Modified
randy-chromium.pcapng
Networking Secure Protocols v0.5
ssl-key.log
File name: randy-chromium.pcapng
Open
Cancel
Help
Files of type: All Capture Files
Automatically detect file type
Format: Wireshark/... - pcapng
Size: 213 KiB, 468 data record(s)
Start / elapsed: 2024-09-12 15:11:49 / 00:00:52
Donate
Networking Secure Protocols v0.5
1h 53min 15s

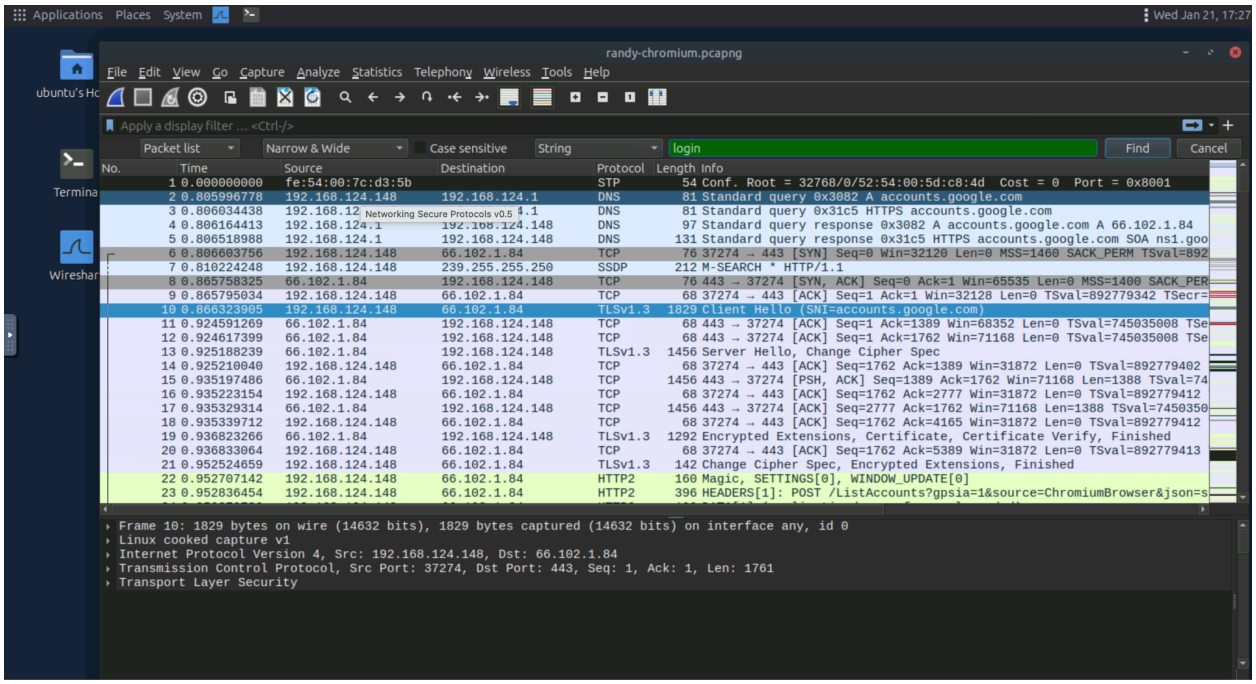

Applications Places System
Wed Jan 21, 17:26
randy-chromium.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 fe:54:00:7c:d3:5b STP 54 Conf. Root = 32768/0/52:54:00:5d:c8:4d Cost = 0 Port = 0x8001
2 0.805996778 192.168.124.148 192.168.124.1 DNS 81 Standard query 0x3082 A accounts.google.com
3 0.806034438 192.168.124.148 192.168.124.1 DNS 81 Standard query 0x31c5 HTTPS accounts.google.com
4 0.806164413 192.168.124.1 192.168.124.148 DNS 97 Standard query response 0x3082 A accounts.google.com A 66.102.1.84
5 0.806518988 192.168.124.1 192.168.124.148 DNS 131 Standard query response 0x31c5 HTTPS accounts.google.com SOA ns1.googl
6 0.806603756 192.168.124.148 66.102.1.84 TCP 76 37274 → 443 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=89277
7 0.810224248 192.168.124.148 239.255.255.250 SSDP 212 M-SEARCH * HTTP/1.1
8 0.865758325 66.102.1.84 192.168.124.148 TCP 76 443 → 37274 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM
9 0.865795034 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=892779342 TSecr=74
10 0.866323905 TLSv1.3 1829 Client Hello (SNI=accounts.google.com)
11 0.924591269 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1389 Win=68352 Len=0 TSval=745035008 TSecr
12 0.924617399 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1762 Win=71168 Len=0 TSval=745035008 TSecr
13 0.925188239 TLSv1.3 1456 Server Hello, Change Cipher Spec
14 0.925210040 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=1389 Win=31872 Len=0 TSval=892779402 TS
15 0.935197486 TCP 1456 443 → 37274 [PSH, ACK] Seq=1389 Ack=1762 Win=71168 Len=1388 TSval=7450
16 0.935223154 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=2777 Win=31872 Len=0 TSval=892779412 TS
17 0.935329314 TCP 1456 443 → 37274 [ACK] Seq=2777 Ack=1762 Win=71168 Len=1388 TSval=745035008
18 0.935339712 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=4165 Win=31872 Len=0 TSval=892779412 TS
19 0.936823266 TLSv1.3 1292 Application Data
20 0.936833064 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=5389 Win=31872 Len=0 TSval=892779413 TS
21 0.952524659 TLSv1.3 142 Change Cipher Spec, Application Data
22 0.952707142 TLSv1.3 160 Application Data
23 0.952836454 TLSv1.3 396 Application Data
24 0.952879736 TLSv1.3 100 Application Data
Mark/Unmark Packet(s) Ctrl+M
Ignore/Unignore Packet(s) Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window
Frame
Internet Protocol Version 4
Transmission Control Protocol
Transport Layer Security
Open Transport Layer Security preferences…
RSA keys list…
Networking Secure Protocols v0.5
TLS debug file: …
Reassemble TLS records spanning multiple TCP segments
Reassemble TLS Application Data spanning multiple TLS records
Message Authentication Code (MAC), ignore "mac failed"
Pre-Shared Key: …
(Pre)-Master-Secret log filename: …
Disable TLS
Networking Secure Protocols v0.5
1h 51min 33s

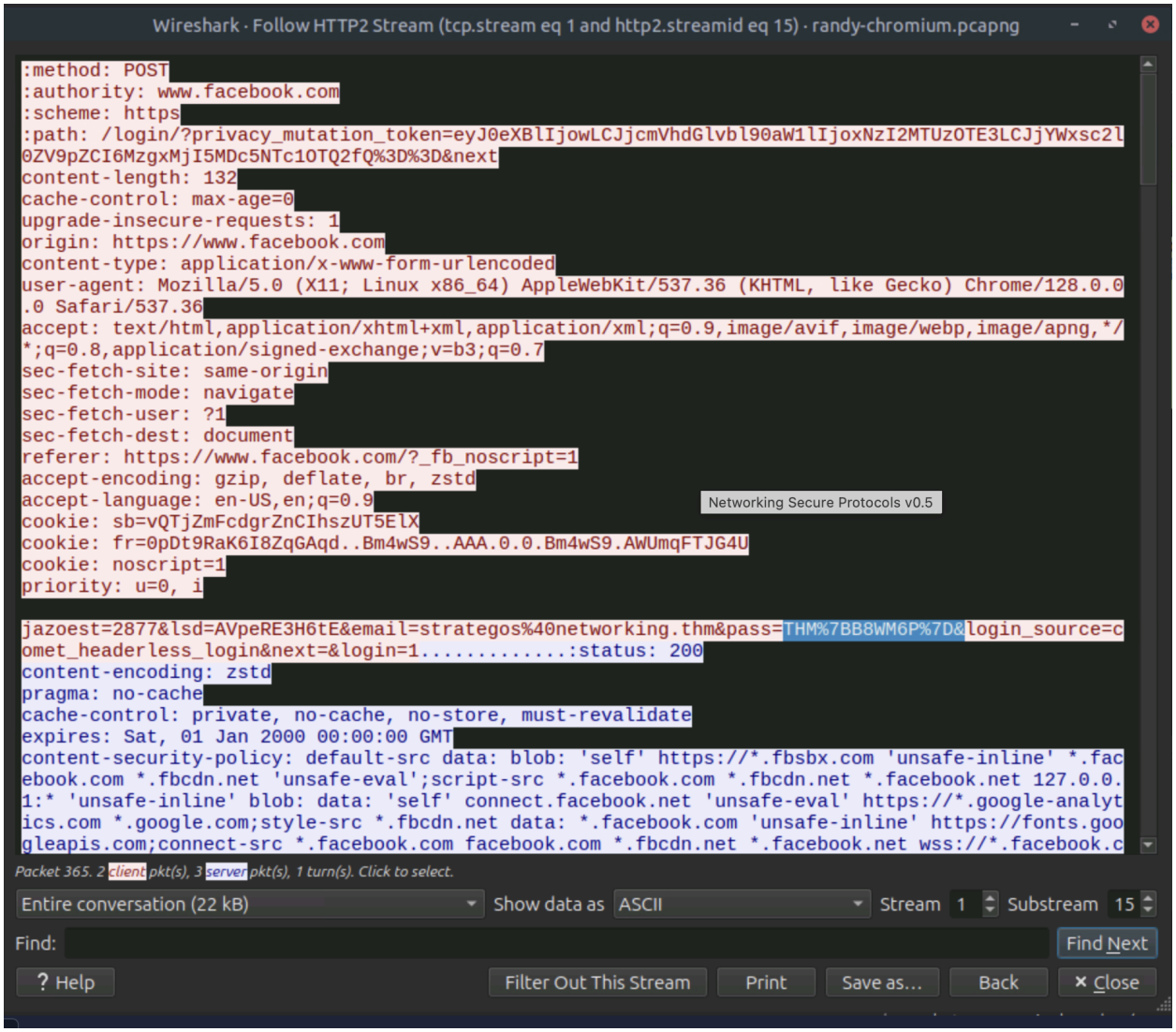


Ctrl + f choose string and enter login




Applications Places System
Wed Jan 21, 17:27
randy-chromium.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Packet list
Narrow & Wide
Case sensitive
String
login
Find
Cancel
No. Time Source Destination Protocol Length Info
1 0.000000000 fe:54:00:7c:d3:5b STP 54 Conf. Root = 32768/0/52:54:00:5d:c8:4d Cost = 0 Port = 0x8001
2 0.805996778 192.168.124.148 192.168.124.1 DNS 81 Standard query 0x3082 A accounts.google.com
3 0.806034438 192.168.12 4.1 DNS 81 Standard query 0x31c5 HTTPS accounts.google.com
4 0.806164413 192.168.124.1 192.168.124.148 DNS 97 Standard query response 0x3082 A accounts.google.com A 66.102.1.84
5 0.806518988 192.168.124.1 192.168.124.148 DNS 131 Standard query response 0x31c5 HTTPS accounts.google.com SOA ns1.goo
6 0.806603756 192.168.124.148 66.102.1.84 TCP 76 37274 → 443 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=892
7 0.810224248 192.168.124.148 239.255.255.250 SSDP 212 M-SEARCH * HTTP/1.1
8 0.865758325 66.102.1.84 192.168.124.148 TCP 76 443 → 37274 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PER
9 0.865795034 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=892779342 TSecr=
10 0.866323905 192.168.124.148 66.102.1.84 TLSv1.3 1829 Client Hello (SNI=accounts.google.com)
11 0.924591269 66.102.1.84 192.168.124.148 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1389 Win=68352 Len=0 TSval=745035008 TSe
12 0.924617399 66.102.1.84 192.168.124.148 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1762 Win=71168 Len=0 TSval=745035008 TSe
13 0.925188239 66.102.1.84 192.168.124.148 TLSv1.3 1456 Server Hello, Change Cipher Spec
14 0.925210040 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=1389 Win=31872 Len=0 TSval=892779402
15 0.935197486 66.102.1.84 192.168.124.148 TCP 1456 443 → 37274 [PSH, ACK] Seq=1389 Ack=1762 Win=71168 Len=1388 TSval=74
16 0.935223154 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=2777 Win=31872 Len=0 TSval=892779412
17 0.935329314 66.102.1.84 192.168.124.148 TCP 1456 443 → 37274 [ACK] Seq=2777 Ack=1762 Win=71168 Len=1388 TSval=7450350
18 0.935339712 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=4165 Win=31872 Len=0 TSval=892779412
19 0.936823266 66.102.1.84 192.168.124.148 TLSv1.3 1292 Encrypted Extensions, Certificate, Certificate Verify, Finished
20 0.936833064 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=5389 Win=31872 Len=0 TSval=892779413
21 0.952524659 192.168.124.148 66.102.1.84 TLSv1.3 142 Change Cipher Spec, Encrypted Extensions, Finished
22 0.952707142 192.168.124.148 66.102.1.84 HTTP2 160 Magic, SETTINGS[0], WINDOW_UPDATE[0]
23 0.952836454 192.168.124.148 66.102.1.84 HTTP2 396 HEADERS[1]: POST /ListAccounts?gpsia=1&source=ChromiumBrowser&json=s
Networking Secure Protocols v0.5
Frame 10: 1829 bytes on wire (14632 bits), 1829 bytes captured (14632 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.124.148, Dst: 66.102.1.84
Transmission Control Protocol, Src Port: 37274, Dst Port: 443, Seq: 1, Ack: 1, Len: 1761
Transport Layer Security


Applications Places System
Wed Jan 21, 17:27
randy-chromium.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Packet list
Narrow & Wide
Case sensitive
String
login
Find
Cancel
No. Time Source Destination Protocol Length Info
1 0.000000000 fe:54:00:7c:d3:5b STP 54 Conf. Root = 32768/0/52:54:00:5d:c8:4d Cost = 0 Port = 0x8001
2 0.805996778 192.168.124.148 192.168.124.1 DNS 81 Standard query 0x3082 A accounts.google.com
3 0.806034438 192.168.12 4.1 DNS 81 Standard query 0x31c5 HTTPS accounts.google.com
4 0.806164413 192.168.124.1 192.168.124.148 DNS 97 Standard query response 0x3082 A accounts.google.com A 66.102.1.84
5 0.806518988 192.168.124.1 192.168.124.148 DNS 131 Standard query response 0x31c5 HTTPS accounts.google.com SOA ns1.goo
6 0.806603756 192.168.124.148 66.102.1.84 TCP 76 37274 → 443 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=892
7 0.810224248 192.168.124.148 239.255.255.250 SSDP 212 M-SEARCH * HTTP/1.1
8 0.865758325 66.102.1.84 192.168.124.148 TCP 76 443 → 37274 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PER
9 0.865795034 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=892779342 TSecr=
10 0.866323905 192.168.124.148 66.102.1.84 TLSv1.3 1829 Client Hello (SNI=accounts.google.com)
11 0.924591269 66.102.1.84 192.168.124.148 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1389 Win=68352 Len=0 TSval=745035008 TSe
12 0.924617399 66.102.1.84 192.168.124.148 TCP 68 443 → 37274 [ACK] Seq=1 Ack=1762 Win=71168 Len=0 TSval=745035008 TSe
13 0.925188239 66.102.1.84 192.168.124.148 TLSv1.3 1456 Server Hello, Change Cipher Spec
14 0.925210040 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=1389 Win=31872 Len=0 TSval=892779402
15 0.935197486 66.102.1.84 192.168.124.148 TCP 1456 443 → 37274 [PSH, ACK] Seq=1389 Ack=1762 Win=71168 Len=1388 TSval=74
16 0.935223154 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=2777 Win=31872 Len=0 TSval=892779412
17 0.935329314 66.102.1.84 192.168.124.148 TCP 1456 443 → 37274 [ACK] Seq=2777 Ack=1762 Win=71168 Len=1388 TSval=7450350
18 0.935339712 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=4165 Win=31872 Len=0 TSval=892779412
19 0.936823266 66.102.1.84 192.168.124.148 TLSv1.3 1292 Encrypted Extensions, Certificate, Certificate Verify, Finished
20 0.936833064 192.168.124.148 66.102.1.84 TCP 68 37274 → 443 [ACK] Seq=1762 Ack=5389 Win=31872 Len=0 TSval=892779413
21 0.952524659 192.168.124.148 66.102.1.84 TLSv1.3 142 Change Cipher Spec, Encrypted Extensions, Finished
22 0.952707142 192.168.124.148 66.102.1.84 HTTP2 160 Magic, SETTINGS[0], WINDOW_UPDATE[0]
23 0.952836454 192.168.124.148 66.102.1.84 HTTP2 396 HEADERS[1]: POST /ListAccounts?gpsia=1&source=ChromiumBrowser&json=s
Networking Secure Protocols v0.5
Frame 10: 1829 bytes on wire (14632 bits), 1829 bytes captured (14632 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.124.148, Dst: 66.102.1.84
Transmission Control Protocol, Src Port: 37274, Dst Port: 443, Seq: 1, Ack: 1, Len: 1761
Transport Layer Security


Wireshark · Follow HTTP2 Stream (tcp.stream eq 1 and http2.streamid eq 15) · randy-chromium.pcapng
:method: POST
:authority: www.facebook.com
:scheme: https
:path: /login/?privacy_mutation_token=eyJ0eXBlIjowLCJjcmVhdGlvbl90aW1lIjoxNzI2MTUzOTE3LCJjYWxsc2l
0ZV9pZCI6MzgxMjI5MDc5NTc1OTQ2fQ%3D%3D&next
content-length: 132
cache-control: max-age=0
upgrade-insecure-requests: 1
origin: https://www.facebook.com
content-type: application/x-www-form-urlencoded
user-agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0
.0 Safari/537.36
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.7
sec-fetch-site: same-origin
sec-fetch-mode: navigate
sec-fetch-user: ?1
sec-fetch-dest: document
referer: https://www.facebook.com/?_fb_noscript=1
accept-encoding: gzip, deflate, br, zstd
accept-language: en-US,en;q=0.9
Networking Secure Protocols v0.5
cookie: sb=vQTjZmFcdgrZnCIhszUT5ElX
cookie: fr=0pDt9RaK6I8ZqGAqd..Bm4wS9..AAA.0.0.Bm4wS9.AWUmqFTJG4U
cookie: noscript=1
priority: u=0, i
jazoest=2877&lsd=AVpeRE3H6tE&email=strategos%40networking.thm&pass=THM%7BB8WM6P%7D&login_source=c
omet_headerless_login&next=&login=1.............:status: 200
content-encoding: zstd
pragma: no-cache
cache-control: private, no-cache, no-store, must-revalidate
expires: Sat, 01 Jan 2000 00:00:00 GMT
content-security-policy: default-src data: blob: 'self' https://*.fbsbx.com 'unsafe-inline' *.fac
ebook.com *.fbcdn.net 'unsafe-eval';script-src *.facebook.com *.fbcdn.net *.facebook.net 127.0.0.
1:* 'unsafe-inline' blob: data: 'self' connect.facebook.net 'unsafe-eval' https://*.google-analyt
ics.com *.google.com;style-src *.fbcdn.net data: *.facebook.com 'unsafe-inline' https://fonts.goo
gleapis.com;connect-src *.facebook.com facebook.com *.fbcdn.net *.facebook.net wss://*.facebook.c
Packet 365. 2 client pkt(s), 3 server pkt(s), 1 turn(s). Click to select.
Entire conversation (22 kB)
Show data as
ASCII
Stream
1
Substream
15
Find:
Find Next
? Help
Filter Out This Stream
Print
Save as...
Back
× Close