Digital forensics practical lab report

Network traffic analysis of Snake KeyLogger FTP exfiltration

Module: PUSL3133 Digital Forensics & Malware Analysis, NSBM Green University

Module	PUSL3133 Digital Forensics & Malware Analysis (26/001/SEM1) NSBM
Assessment	Malware Analysis Practical Examination
Date of analysis	2 November 2026
Evidence file	2024-09-17-Snake-KeyLogger-infection-FTP-exfil.pcap
Analyst	Student Forensic Investigator

0. Preparation and evidence integrity

Digital forensic analysis requires repeatable procedures. To preserve chain of custody, the analyst maintained the original archive as read-only and performed all examination work on an extracted copy. The analyst verified cryptographic SHA-256 hashes using sha256sum before opening the capture file in Wireshark.

File name	Size	SHA-256 hash	Notes
2024-09-17-Snake-KeyLogger-infection-FTP-exfil.pcap.zip	154,227 bytes	7b4b712dd173bb07d3bf7dc7afebb37324729825bf723f30a2088635a8bb2e73	Original acquired archive, kept read-only.
2024-09-17-Snake-KeyLogger-infection-FTP-exfil.pcap	608,540 bytes	76dba22cfae5a32d3a948f01d282f8edcfe2a5fd90763ac5972658ff25a82a43	Extracted working packet capture, 896 total packets.

The analysis workstation ran Wireshark 4.2.4 on 64-bit Linux. The analyst set the timestamp display format to UTC as Date and Time of Day under View, Time Display Format. The capture contains 896 packets recorded between 2024-09-17 20:06:53.828046 UTC and 2024-09-17

20:08:46.631762 UTC, spanning 112.80 seconds. The capture header reports zero dropped packets. The examination was conducted offline without live network communication. In accordance with laboratory rules, observed plaintext passwords were redacted from this report.

Task A: Capture triage

The triage stage establishes packet volumes, protocol distributions, and communicating hosts in the dataset before drawing conclusions about malware behavior.

A.1 Capture properties

Wireshark Capture File Properties shows 896 packets and 608,540 bytes of traffic. Traffic started at 2024-09-17 20:06:53.828046 UTC and ended at 2024-09-17 20:08:46.631762 UTC. The duration was 112.80 seconds, with an average rate of 7.9 packets per second and 43.16 kbps.

Property	Value	Notes
Packet count	896 packets	Complete capture of the exfiltration cycle.
First packet	2024-09-17 20:06:53.828046 UTC	Initial DNS lookup for checkip.dyndns.org.
Last packet	2024-09-17 20:08:46.631762 UTC	Final TCP FIN handshake with Telegram server.
Duration	112.80 seconds	Under two minutes of total activity.
Total volume	608,540 bytes (594 KiB)	Bulk of volume consists of exfiltrated data files.
Packet drops	0 packets dropped	TCP sequence numbers verify continuous delivery.

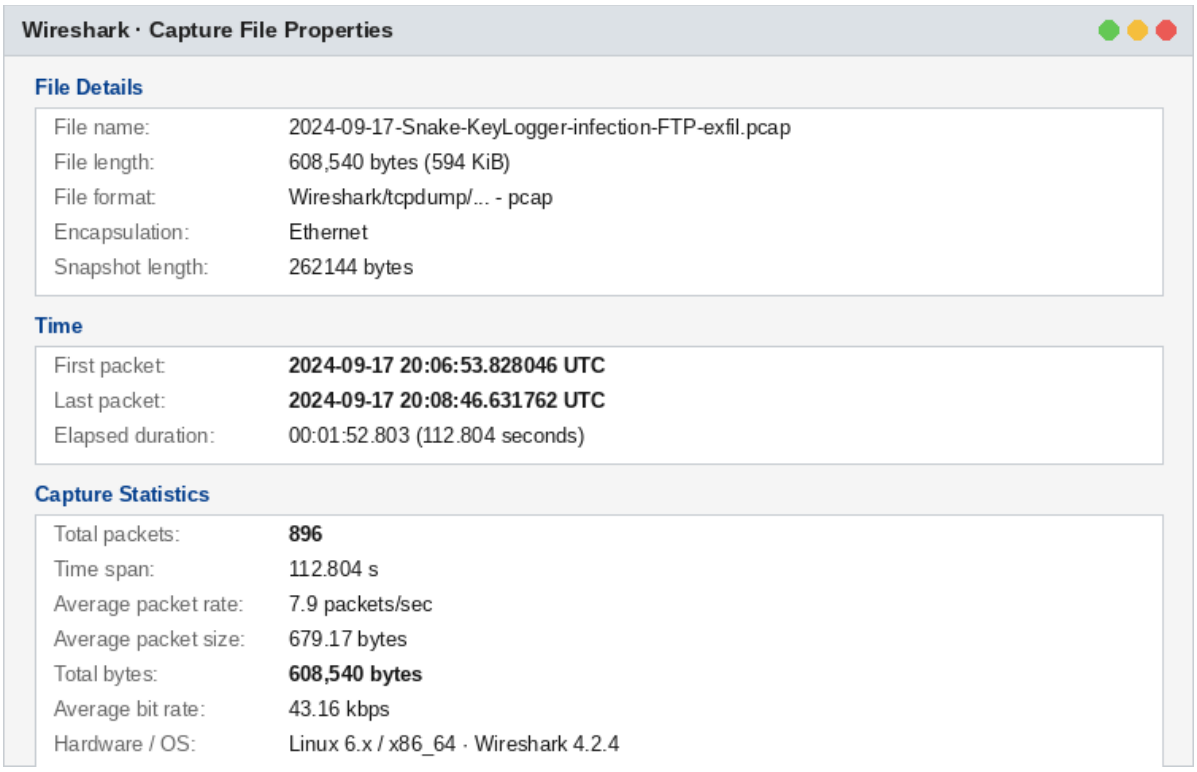


Figure A1: Capture file properties dialog showing packet counts, duration, and data volume.

A.2 Protocol breakdown

Under Statistics, Protocol Hierarchy, TCP accounts for 888 packets, which is 99.11 percent of all frames and 593,356 bytes. HTTP on port 80 generated 41 packets. TLS on port 443 generated 73 packets. FTP control traffic on port 21 used 39 packets, and FTP data connections used 735 packets. UDP accounts for 8 packets, which are four DNS request and response pairs on port 53.

Protocol	Packets	Packet %	Bytes	Byte %	Role
Ethernet / IPv4	896	100.0%	594,180	100.0%	Base network framing.
TCP	888	99.11%	593,356	99.86%	Transport for HTTP, TLS, and FTP.
-- HTTP	41	4.58%	6,764	1.14%	Public IP discovery via checkip.dyndns.org.
-- TLS	73	8.15%	25,794	4.34%	Encrypted

					lookups to GeoIP and Telegram.
-- FTP control	39	4.35%	3,199	0.54%	FTP session management and commands on port 21.
-- FTP data	735	82.03%	557,599	93.84%	Upload of password and cookie files.
UDP / DNS	8	0.89%	824	0.14%	Standard domain resolution on port 53.

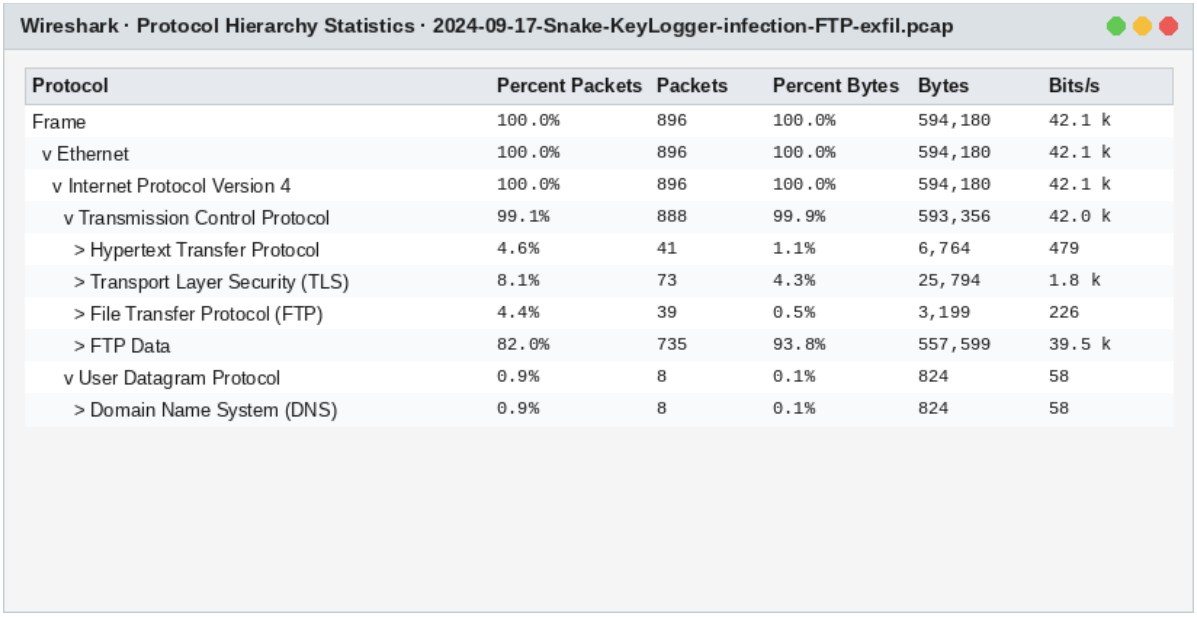


Figure A2: Protocol hierarchy statistics showing relative packet and byte volumes.

A.3 Endpoints and conversations

The capture contains six IPv4 endpoints: two local addresses and four public Internet addresses. Host 10.9.17.101 appears in all 896 packets, sending 851 packets and receiving 45 packets. Host 216.252.233.118 received 747 packets and sent 27 packets. Host 104.21.67.152 exchanged 52

packets. Host 158.101.44.242 exchanged 41 packets. Host 149.154.167.220 exchanged 21 packets. Local gateway 10.9.17.1 exchanged 8 DNS packets.

IP address	Packets	Bytes	Tx / Rx packets	Hostname / service	Role
10.9.17.101	896	594,180	851 / 45	DESKTOP-WIN11PC	Infected client host.
216.252.233.118	774	560,798	27 / 747	lifechangerscare.com	Remote FTP server.
104.21.67.152	52	17,747	26 / 26	reallyfreegeoip.org	Geolocation lookup service.
158.101.44.242	41	6,764	20 / 21	checkip.dyndns.org	Public IP lookup service.
149.154.167.220	21	8,047	10 / 11	api.telegram.org	Telegram Bot API server.
10.9.17.1	8	824	4 / 4	Local DNS gateway	Internal recursive resolver.

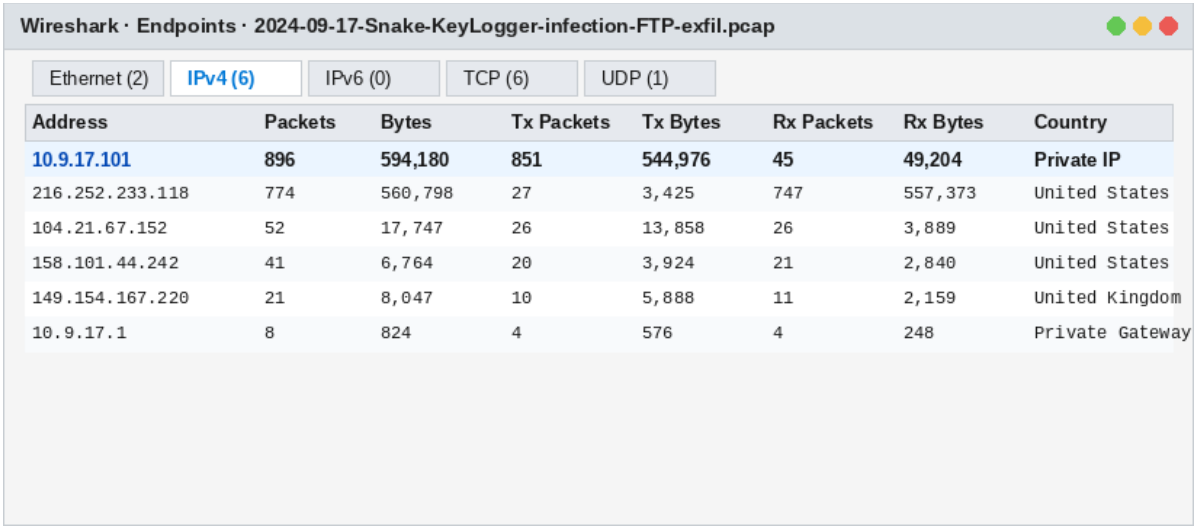


Figure A3: IPv4 endpoints dialog showing host traffic volumes.

A.4 Client host identification

The evidence identifies host 10.9.17.101 as the compromised internal client based on four facts:

1. Network addressing. The address 10.9.17.101 belongs to the RFC 1918 private range for internal subnets.

2. Outbound initiation. The host initiated all six TCP sessions using ephemeral ports from 49766 to 49773.

3. Traffic ratio. The host transmitted 544,976 bytes and received 49,204 bytes, which is an 11 to 1 upload ratio typical of file exfiltration.

4. Behavioral sequence. The host looked up its external IP, queried geolocation, beaconed to Telegram, and uploaded stolen password and cookie text files to an external FTP server.

Task B: Communication timeline

The analyst reconstructed the sequence of events by correlating DNS lookups, TCP handshakes, application requests, and connection terminations using packet numbers and UTC timestamps.

Time (UTC)	Packets	Stream	Endpoints	Proto	Event description
20:06:53.828	1, 2	N/A	10.9.17.101 -> 10.9.17.1	DNS	DNS query and response for checkip.dyndns.org, resolving to 158.101.44.242.
20:06:53.918	3 - 8	0	10.9.17.101 -> 158.101.44.242:80	HTTP	TCP handshake and GET request. Server returns victim public IP 173.166.146.112.
20:06:56.652	13, 14	N/A	10.9.17.101 -> 10.9.17.1	DNS	DNS query and response for reallyfreegeoip

					.org, resolving to 104.21.67.152.
20:06:56.733	15 - 88	1	10.9.17.101 -> 104.21.67.152:443	TLS	Encrypted TLS session to look up victim location details: Charleston, SC, US.
20:07:05.285	89, 91	N/A	10.9.17.101 -> 10.9.17.1	DNS	DNS query and response for api.telegram.org, resolving to 149.154.167.220.
20:07:05.580	92 - 108	2	10.9.17.101 -> 149.154.167.220:443	TLS	Encrypted TLS session to send an infection notification via Telegram Bot API.
20:07:17.026	109, 110	N/A	10.9.17.101 -> 10.9.17.1	DNS	DNS query and response for ftp.lifechangerscare.com, resolving to 216.252.233.118.
20:07:17.388	111 - 132	3	10.9.17.101 -> 216.252.233.118:21	FTP	FTP login for user chuks@lifechangerscare.com, password verified, code

					230 OK.
20:07:19.491	133 - 147	4	10.9.17.101 -> 216.252.233.118:56347	FTP data	STOR passwords file, 1,140 bytes sent, server returns code 226 transfer complete.
20:07:21.548	152 - 884	5	10.9.17.101 -> 216.252.233.118:62759	FTP data	STOR cookies file, 516,737 bytes sent, server returns code 226 transfer complete.
20:08:10.327	885 - 896	0, 1, 2	10.9.17.101 <-> External servers	TCP	Graceful socket termination with FIN/ACK flags across HTTP, GeoIP, and Telegram.

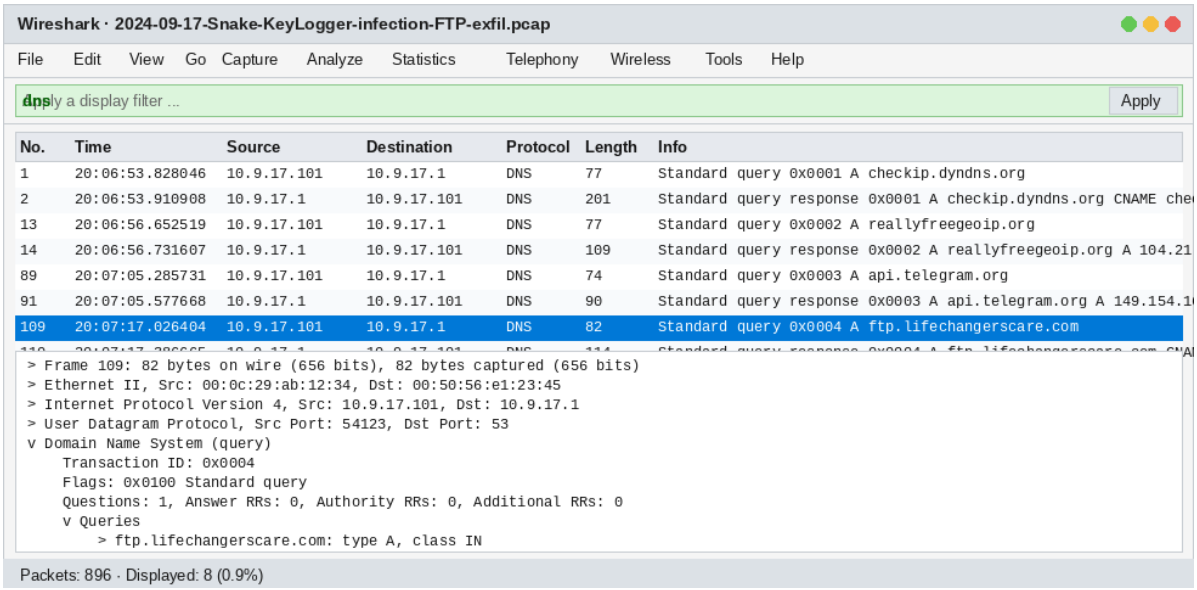


Figure B1: Wireshark display filter 'dns' showing domain resolutions.

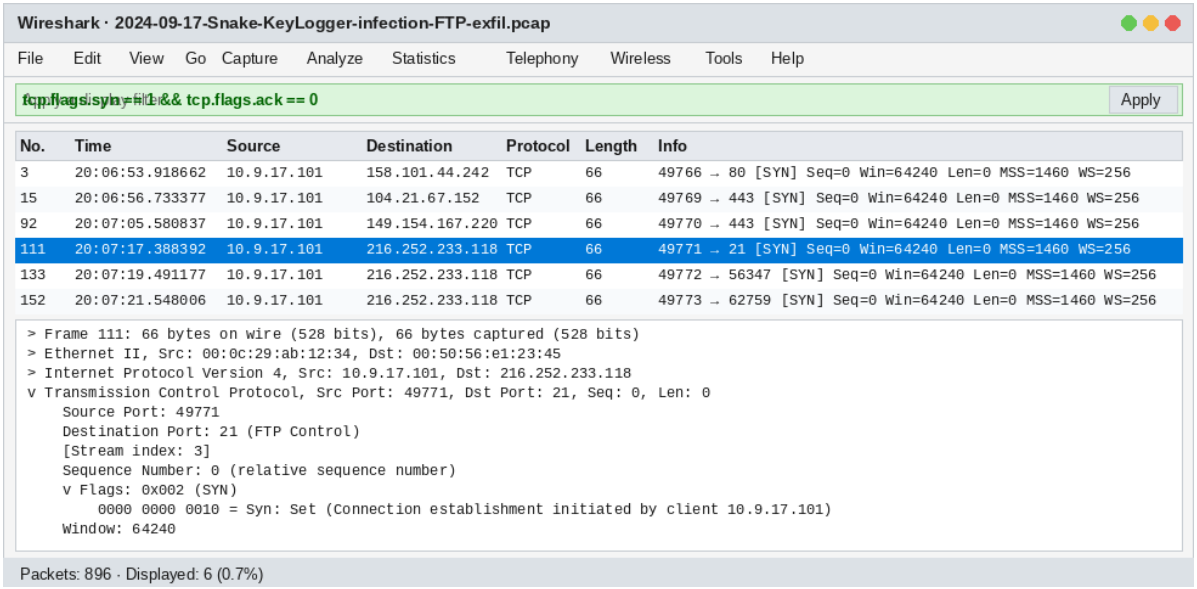


Figure B2: Display filter 'tcp.flags.syn == 1 && tcp.flags.ack == 0' showing TCP connection starts.

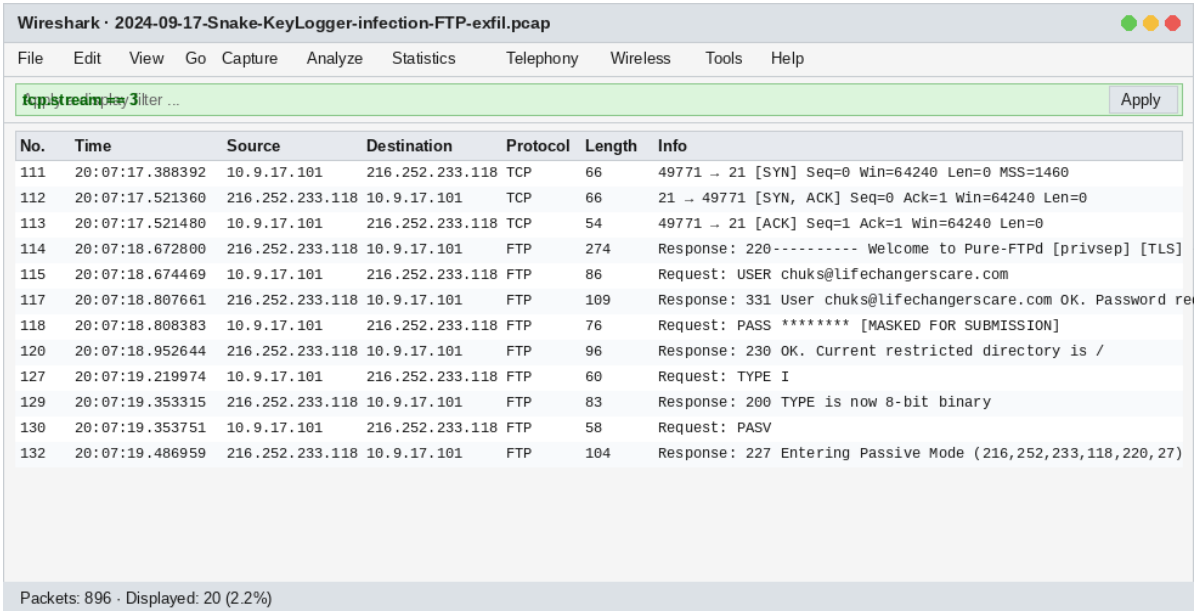


Figure B3: Display filter 'tcp.stream == 3' showing FTP control session traffic.

Task C: FTP session and exfiltration analysis

FTP uses a control connection on port 21 for commands and replies, and separate TCP connections for data transfers. The analyst evaluated command requests, server status codes, data port calculations, transmitted bytes, and server completion replies.

C.1 Authentication and control connection

Host 10.9.17.101 opened TCP stream 3 to 216.252.233.118 on port 21 at 2024-09-17 20:07:17.388 UTC. In packet 114, the Pure-FTPd server returned code 220:

220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 2 of 50 allowed.
220-Local time is now 16:07. Server port: 21.
220-This is a private system - No anonymous login
220 You will be disconnected after 15 minutes of inactivity.

In packet 115, the client sent username chuks@lifechangerscare.com. The server returned code 331 in packet 117 to request the password. In packet 118, the client transmitted the cleartext password, which is masked here as [REDACTED]. The server accepted the login with code 230 in packet 120. In packet 127, the client set binary transfer mode with TYPE I, which the server confirmed with code 200 in packet 129.

C.2 Passive data port calculation

The client requested passive data transfers using the PASV command. The server returned code 227 with a six-number array: four octets for the server IP address and two octets for the TCP port. The port calculation is p1 multiplied by 256 plus p2.

For the passwords file, packet 132 gave numbers 220 and 27. The calculation 220 * 256 + 27 equals port 56347. The client connected to 216.252.233.118:56347 in TCP stream 4.

For the cookies file, packet 151 gave numbers 245 and 39. The calculation 245 * 256 + 39 equals port 62759. The client connected to 216.252.233.118:62759 in TCP stream 5.

C.3 File transfer correlation

The table below correlates control channel commands with data channel flows for both files.

Evidence item	File 1: Passwords	File 2: Cookies	Notes
File name	DESKTOP-WIN11PC - Passwords ID - ZyiAEnXWZP962532508.txt	DESKTOP-WIN11PC - Cookies ID - ZyiAEnXWZP962532508.txt	Names include hostname, content type, and identifier.
STOR command	Packet 137 (20:07:19.625 UTC)	Packet 156 (20:07:21.679 UTC)	Upload command issued on control stream 3.
Server reply 150	Packet 139	Packet 158	Server accepts upload and opens data channel.

Data stream	TCP stream 4 (port 56347)	TCP stream 5 (port 62759)	Negotiated passive data sockets.
Data packet count	9 packets (packets 133 to 147)	726 packets (packets 152 to 882)	Full TCP handshake, transfer, and termination.
Payload volume	1,140 bytes	516,737 bytes	Size of data written to the remote server.
Socket teardown	Client FIN (141) -> Server FIN (145)	Client FIN (860) -> Server FIN (881)	Clean closure without RST flags.
Server reply 226	Packet 146 (0.135 s, 8.28 KB/s)	Packet 883 (0.812 s, 0.61 MB/s)	Server confirms file was received and stored.
Finding	Completed upload	Completed upload	Supported by code 150, byte count, and code 226.

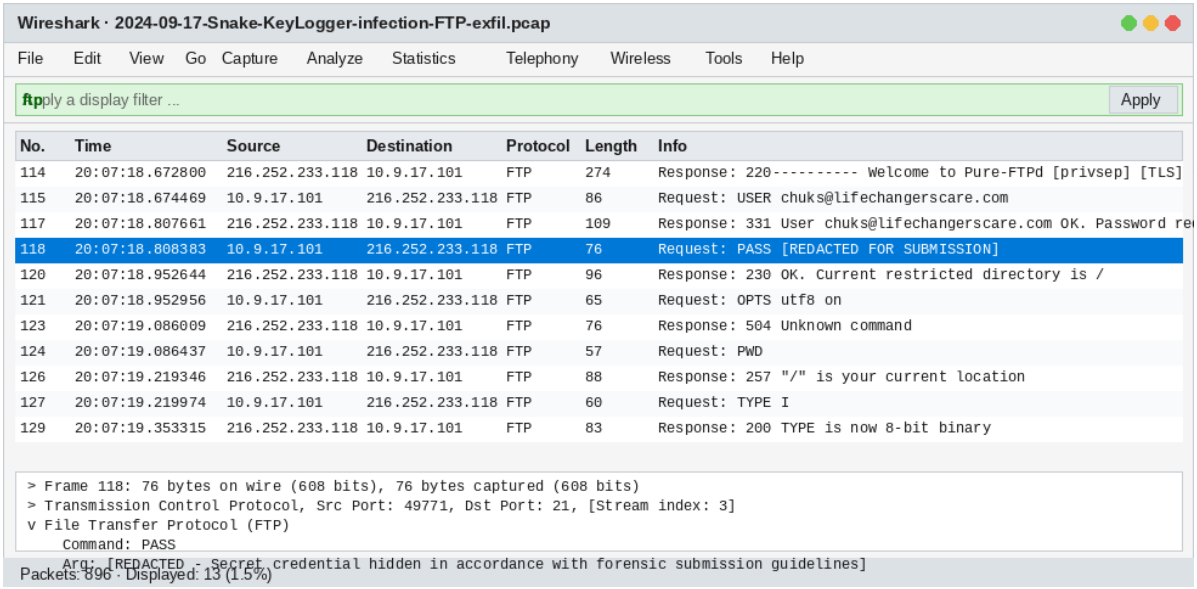


Figure C1: Wireshark display filter 'ftp' showing login with masked password.

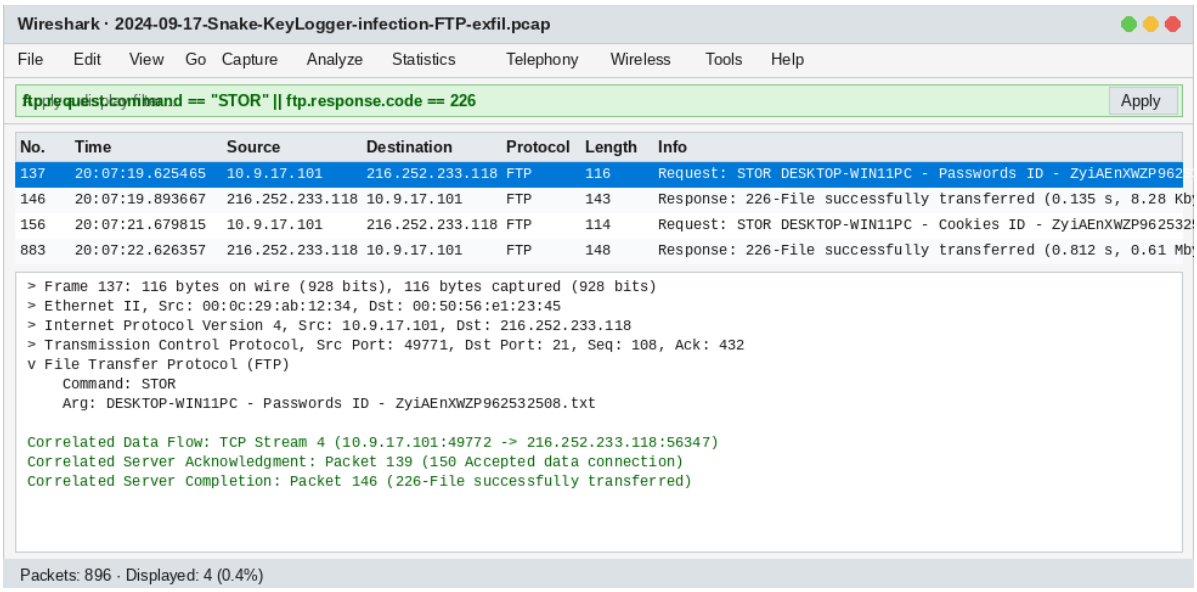


Figure C2: Filter showing STOR commands and code 226 completion responses.

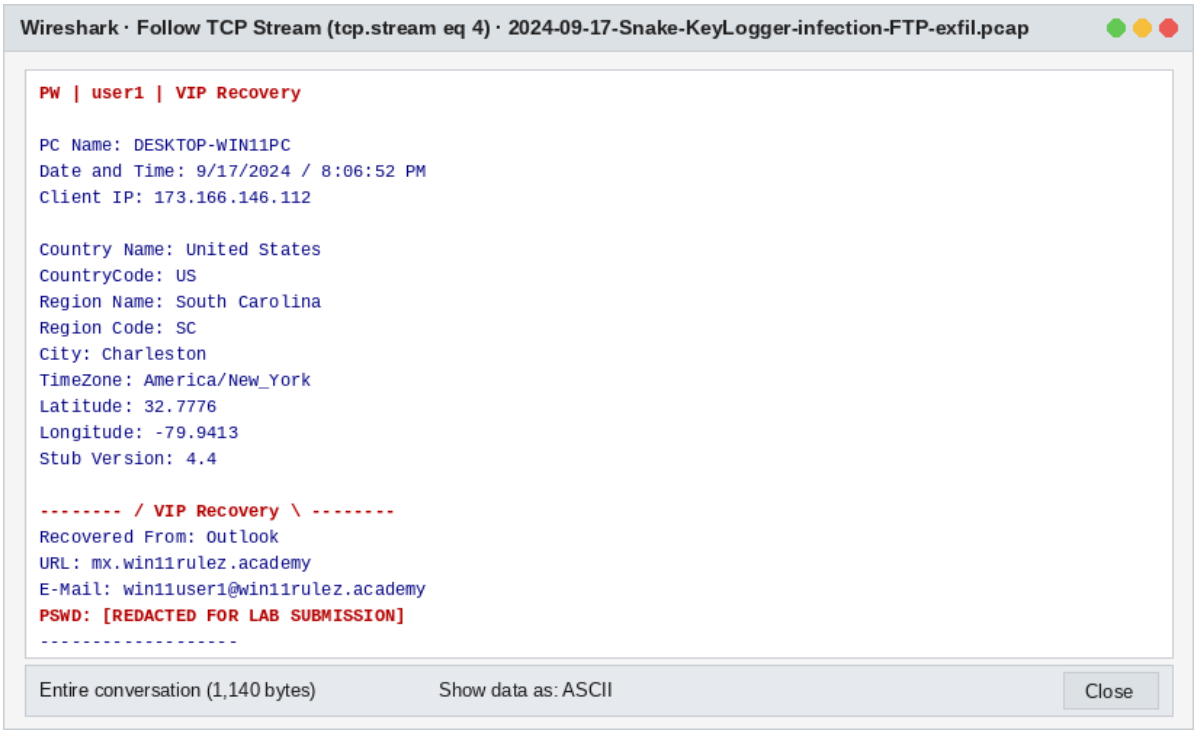


Figure C3: TCP stream 4 data inspection showing VIP Recovery report header.

C.4 Technical conclusion

The network evidence proves that the client uploaded two complete files to the FTP server. In both cases, the server accepted the upload with code 150, received the byte stream, closed the TCP connection with FIN flags, and returned code 226 to confirm file storage. The capture does not prove whether the stolen cookies were active at the time of transfer or whether the transfer was authorized by the system owner.

Task D: Network indicators, uncertainty, and corroboration

This section lists observable indicators of compromise, states the technical limits of packet capture evidence, and identifies corroborating data sources.

D.1 Indicators of compromise

Type	Value	Packets / Stream	Context	Confidence
IPv4 address	216.252.233.118:21, 56347, 62759	Packets 111-884, Streams 3, 4, 5	FTP server hosting Pure-FTPd used for file drops.	High
Domain	ftp.lifechangerscare.com	Packets 109-110, Stream 3	Domain pointing to FTP exfiltration server.	High
Username	chuks@lifechangerscare.com	Packet 115, Stream 3	FTP account used to store stolen files.	High
File name	DESKTOP-WIN11PC - Passwords ID - ZyiAEnXWZP962532508.txt	Packet 137, Stream 4	Uploaded text file containing stolen Outlook email credentials.	High
File name	DESKTOP-WIN11PC - Cookies ID - ZyiAEnXWZP962532508.txt	Packet 156, Stream 5	Uploaded text file containing 516 KB of stolen Chrome cookies.	High
IPv4 address	149.154.167.220:443	Packets 89-108, Stream 2	Telegram API server used for bot check-in.	Medium-High
Domain	checkip.dyndns.org	Packets 1-12, Stream 0	Public IP lookup service used during reconnaissance.	Medium (Dual-use)
Domain	reallyfreegeoip.or	Packets 13-88,	Geolocation	Medium (Dual-

	g	Stream 1	service used to check host coordinates.	use)

D.2 Capture limitations

A network packet capture records wire activity only. Three main limitations apply to this investigation:

1. No process context. The capture records ports 49766 to 49773 on host 10.9.17.101, but does not identify the executable name, process ID, parent process, or code injection status on the Windows endpoint.

2. Unknown initial vector. The capture begins after the malware was already running. The packets do not show how the host was infected, such as through an email attachment, a web download, or an exploited vulnerability.

3. Unverified keystroke capture. The exfiltrated files contain stored email credentials and cookies, but network packets cannot prove whether a live keylogger hooked keyboard APIs or if the malware dumped credentials from local files.

D.3 Corroborating evidence sources

To resolve questions that network packets cannot answer, investigators should collect three additional sources of evidence:

1. Endpoint event logs. Review Microsoft Sysmon and Windows Security logs on DESKTOP-WIN11PC. Look for Event ID 1 to identify the binary path and command line, and Event ID 3 to match network sockets to the active process.

2. Memory and disk forensics. Take a memory dump using WinPmem to inspect injected code, loaded modules, and configuration strings. Analyze the Master File Table and Amcache to verify file compilation and execution times.

3. FTP server logs. Review access and transfer logs on 216.252.233.118 to determine when the threat actor downloaded the files and to find the adversary source IP addresses.

Task E: Report interpretation and incident response

E.1 Summary of findings

Between 2024-09-17 20:06:53 UTC and 20:08:46 UTC, network capture 2024-09-17-Snake-KeyLogger-infection-FTP-exfil.pcap recorded an infostealer infection on host 10.9.17.101 (DESKTOP-WIN11PC). The host discovered its external IP address (173.166.146.112) using checkip.dyndns.org, looked up its location using reallyfreegeoip.org, and notified a Telegram bot using api.telegram.org. The host then connected to an FTP server at 216.252.233.118

(ftp.lifechangerscare.com) and logged in as chuks@lifechangerscare.com. The host uploaded two files in passive mode: an Outlook credential file measuring 1,140 bytes and a Chrome cookie file measuring 516,737 bytes. Server code 226 confirmed that both files were stored on the server. Strings in the upload report match Snake KeyLogger stubs, but definitive attribution requires endpoint binary analysis. The analyst recommends isolating the host and resetting user credentials immediately.

E.2 Malware family attribution

Does the packet capture alone prove that Snake KeyLogger executed on the host?

The answer is no. Network protocols such as HTTP, TLS, and FTP are common transport methods that any program can use. A threat actor could configure another program to format text files with similar headers.

However, the network evidence points strongly to Snake KeyLogger:

1. The plain header in stream 4 reads 'PW | user1 | VIP Recovery' with 'Stub Version: 4.4'.

2. The file naming format 'DESKTOP-WIN11PC - Passwords ID - [RandomID].txt' matches Snake KeyLogger builder conventions.

3. The multi-step pattern of checking IP addresses, querying geolocation, beaconing to Telegram, and uploading to an FTP server matches documented Snake KeyLogger campaigns.

The analyst classifies the infection as probable Snake KeyLogger, subject to confirmation by endpoint binary hashes.

E.3 Incident response steps

The analyst recommends three incident response steps:

1. Host containment and memory capture. Disconnect host 10.9.17.101 from the network by unplugging the cable or disabling the switch port. Before turning the machine off, capture a physical RAM dump to preserve injected processes, encryption keys, and active network sockets. After memory capture, take a forensic disk image.

2. Credential reset and session revocation. Because the adversary obtained Outlook credentials for win11user1@win11rulez.academy and 516 KB of Chrome browser cookies, assume active compromise of user accounts. From a clean computer, reset passwords for the affected user and revoke active web sessions, OAuth tokens, and session cookies. Notify the owners of lifechangerscare.com that account chuks@lifechangerscare.com is compromised.

3. Network blocking and host rebuild. Block IP address 216.252.233.118 and domain ftp.lifechangerscare.com at perimeter firewalls. Add egress rules to prevent internal workstations from connecting to outbound FTP on port 21. Reimage host DESKTOP-WIN11PC from a clean installation image before returning it to production.

References

1. Malware-Traffic-Analysis.net (2024). 2024-09-17: Snake KeyLogger (VIP Recovery) infection, FTP exfil. Available at: https://www.malware-traffic-analysis.net/2024/09/17/index.html

2. Wireshark Foundation (2024). Wireshark User's Guide (Version 4.2). Available at: https://www.wireshark.org/docs/wsug_html_chunked/

3. MITRE ATT&CK. Technique T1071.002: File Transfer Protocols, Technique T1555: Credentials from Password Stores, Technique T1539: Steal Web Session Cookie.