Σχολή Θετικών Επιστημών και Τεχνολογίας

Πρόγραμμα Σπουδών Πληροφορικής

Πτυχιακή Εργασία

Αξιοποίηση δεδομένων TCP/IP Traffic σε Μορφή Γραφήματος Γνώσης για τη Βελτίωση της Κυβερνοασφάλειας

-

Technical Report

Πέτρος-Γιάννης Πεταλάς

Επιβλέπων καθηγητής: Βάιος Παπαϊωάννου

In this report you will find the text generated by Claude, exactly as it was presented. There is a total of 60 conversations. The title parts mean the following:

- MC: Malware Capture
- MX: Mixed Capture
- NC : Normal Capture
- S : Security information present in graph
- NS : No security information in graph

MC-90-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

8:24 PM

Claude responded: I'll analyze the network traffic data in the Neo4j database and cross-reference it with MITRE ATT&CK techniques to identify potential malicious activity.

I'll analyze the network traffic data in the Neo4j database and cross-reference it with MITRE ATT&CK techniques to identify potential malicious activity. Let me start by understanding the database schema.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

Good, I have the schema. Let me explore the data to understand what we're dealing with.

Used neo4j-database integration

Used neo4j-database integration

N

Read Neo4j Cypher

Result

N

Read Neo4j Cypher

Result

Done

Interesting — 16,401 "local" IPs out of 16,406. That's suspicious in itself; in a normal small network there shouldn't be that many local IPs. Let me check the actual IP ranges to understand what "local" really means here.

Used neo4j-database integration

Used neo4j-database integration

All 192.168.x.x — that's classic RFC 1918 private space, but having 16,401 unique private IPs in a /16 is a huge number for a normal capture. Let me look at subnet distribution and conversation patterns.

Read Neo4j Cypher

Read Neo4j Cypher

This is extremely suspicious. The "local" IPs are spread evenly across the entire 192.168.0.0/16 range — roughly 75-80 per /24 subnet across hundreds of subnets. This is

highly indicative of **IP address spoofing** or a **scanning/reflection attack** — not legitimate local network behavior. Let me look at what these IPs are doing.

Read Neo4j Cypher

Read Neo4j Cypher

A very clear pattern: **192.168.3.104** is at the center of everything — either sender or receiver — talking to a huge fan-out of IPs. Let me confirm.

Used neo4j-database integration

Used neo4j-database integration

This is now crystal clear. **192.168.3.104** is involved in 8,473 + 7,938 = **16,411 conversations** — essentially **every conversation in the entire capture**. Every other "local" IP talks to this single host exactly once or twice. This is the textbook signature of either:

1. A host scanning the network (or being scanned)
2. A host acting as a command-and-control hub talking to thousands of distinct peers

Let me look at the ports involved and whether 192.168.3.104 is initiating outbound or receiving inbound.

Read Neo4j Cypher

Read Neo4j Cypher

This is the smoking gun. Port **445 (SMB/Microsoft-DS)** is used in **16,402 out of 16,411 conversations** — essentially every conversation involving 192.168.3.104.

Port 445 SMB activity at this scale, hitting thousands of distinct hosts across an entire /16, is the classic signature of **SMB-based worms and ransomware** — the same propagation method used by **WannaCry, NotPetya, Conficker, EternalBlue-based attacks, and Emotet**. The high-numbered ports (1600s-1900s range) are the dynamic source ports on the other end of those SMB connections.

Let me dig deeper to confirm direction and look at byte volumes.

Used neo4j-database integration

Used neo4j-database integration

Highly uniform packet sizes (288 bytes for 85% of outbound conversations) — that's another classic scanner signature. A real SMB file-sharing session would have hugely variable byte counts. This is a probe.

Used neo4j-database integration

Used neo4j-database integration

Now I see two more interesting things:

1. **192.168.3.104 → 192.168.3.255 (broadcast address)** with 54KB — this is broadcast traffic, possibly NetBIOS name service or SMB browser broadcasts used for host discovery.

2. **192.168.3.104 → 200.10.122.10** with **1.16 MB sent**. That's a substantial outbound transfer to an external IP — possible **data exfiltration** or **C2 callback**.

Let me investigate the external destinations more carefully.

Used neo4j-database integration

Used neo4j-database integration

Now let me check what ports the external traffic used:

Read Neo4j Cypher

Read Neo4j Cypher

Let me also look up the geolocation of those external IPs:

Used neo4j-database integration

Used neo4j-database integration

One conversation per destination — pure scan behavior. Let me now look at MITRE ATT&CK to map this to known techniques.

Searched the web

Searched the web

Excellent — that confirms the MITRE framework mapping. Let me also check the broadcast traffic and whether 192.168.3.104 is the only "active" host or if there are others doing anything:

Used neo4j-database integration

Used neo4j-database integration

Confirmed: 192.168.3.104 is the sole active host — every single conversation in the capture involves it. There is no normal background traffic between any other pair of hosts in this dataset. Combined with the fan-out to 8,473 unique destinations on port 445, this is unambiguous.

Let me also check if any of the destination ports show successful exploitation patterns by looking at byte transfer distributions:

Used neo4j-database integration

Used neo4j-database integration

Now let me verify the MITRE ATT&CK matches by looking up the relevant techniques:

Searched the web

Searched the web

I have enough evidence. Let me also do one final check on the external traffic to better characterize the suspicious outbound transfer:

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a complete picture. Let me build a visualization to summarize the findings before writing the verdict.

Now I have everything I need. Let me create a summary visualization of the attack flow.

Verdict: Strong indications of an active compromise

The capture shows behavior that is **inconsistent with legitimate network use** and matches multiple MITRE ATT&CK techniques. There is one clearly suspicious host in the local network: **192.168.3.104**.

Evidence from the graph

1. Single host dominates 100% of traffic. Every one of the 16,411 conversations in the capture involves 192.168.3.104 as either sender or receiver. Zero traffic flows between any other pair of hosts. In a real working network this is impossible — there should always be background chatter (DNS, DHCP, AD, printing, etc.) between other endpoints. Either the capture was deliberately filtered around this host, or this host is the only one generating traffic worth investigating.

2. Massive fan-out on SMB (port 445). 192.168.3.104 contacted **8,473 distinct IPs across the entire 192.168.0.0/16 range** (75–80 IPs per /24 subnet, evenly spread). Port 445 (Microsoft-DS / SMB) appears in **16,402 of 16,411 conversations** — essentially every probe. **7,937 hosts responded**, meaning the attacker now has a complete inventory of live SMB-enabled machines.

3. Uniform packet sizes. 85% of outbound conversations carry exactly 288 bytes sent / 80 bytes received — the signature of an automated scanner sending the same crafted probe to every target, not real file-sharing sessions.

4. Broadcast discovery. 192.168.3.104 sent ~54 KB to the subnet broadcast address 192.168.3.255 — typical NetBIOS / SMB browser broadcasts used to enumerate the local segment before scanning further.

5. Anomalous outbound transfer. While the SMB scan ran, 192.168.3.104 sent **1.16 MB** to **200.10.122.10** (host10.advance.com.ar, Buenos Aires). The volume is several orders of magnitude larger than every other conversation and goes to an external destination with no obvious business purpose — a classic exfiltration / C2 callback signature.

MITRE ATT&CK mapping

The pattern maps cleanly onto these techniques:

- **T1018 – Remote System Discovery.** Sweeping the /16 to find live hosts. Used by Conti, LadonGo, and many ransomware crews.
- **T1046 – Network Service Discovery.** Port scanning on TCP/445 to enumerate vulnerable SMB services. The textbook precursor to lateral movement and ransomware deployment.

- **T1135 – Network Share Discovery.** SMB enumeration to identify shares for collection and lateral spread. Used by Emotet, QakBot, Ramsay, BlackByte, Medusa, APT39, Tonto Team.
- **T1021.002 – SMB / Windows Admin Shares.** The traffic pattern (one source talking to thousands of targets on 445 that don't normally communicate with it) is the exact behavior the technique's detection guidance flags. Used by WannaCry, NotPetya, Emotet, HermeticWizard.
- **TA0010 – Exfiltration (likely).** The 1.16 MB outbound transfer to Argentina, concurrent with the scan, fits exfiltration or C2 staging.

Caveats

- None of the IPs in the graph were tagged is_known_attacker, is_Bot, or is_spam. The verdict is based on behavior, not reputation.
- "Local" in the schema appears to mean "RFC 1918 private space", not "in our network" — 16,401 distinct private IPs spread across a full /16 is consistent with a scan target list rather than a real host inventory. The actual local network is probably the 192.168.3.0/24 subnet (where 192.168.3.104 and 192.168.3.255 live).
- The capture window appears narrow and centered on this one host's activity. Whether the compromise originated here, or whether 192.168.3.104 is itself a victim being driven from elsewhere, can't be determined from this dataset alone.

Recommendation

Treat **192.168.3.104** as compromised. Isolate it from the network, image it for forensics, audit which of the 7,937 responding hosts have successful SMB authentication events in the same timeframe, and investigate the outbound flow to 200.10.122.10 as a potential exfiltration channel.

MC-90-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

8:59 PM

Claude responded: I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK matrix to identify potential cyber-attack indicators.

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK matrix to identify potential cyber-attack indicators. Let me start by loading the necessary tools and exploring the graph schema.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

I have a clear picture of the graph schema. Now let me explore the data systematically to look for attack indicators.

Used neo4j-database integration

Used neo4j-database integration

Interesting — the schema marks is_Local=true for 16,401 IPs, but only 5 are flagged as non-local. Let me reconcile this with actual addresses:

Used neo4j-database integration

Used neo4j-database integration

Wait — the data marks **all** 192.168.x.x addresses as "Local". Yet there are 16,401 distinct local IPs in a /16 (or smaller), which is unusual for a real LAN. Let me check whether these are actually internal hosts or remote IPs that the dataset has mis-labeled. The capture has 5 actual hosts (66.252.13.214, 72.5.123.29, etc.) which are public IPs.

Let me re-examine — those 5 "Hosts" might be the local network's vantage points (gateways/servers), and the 192.168 addresses are likely the *peers* in conversations:

Read Neo4j Cypher

Read Neo4j Cypher

A **huge red flag** already. One single local IP — 192.168.3.104 — initiates **all 16,411 conversations** as sender, while other IPs only appear as receivers. This is a classic outbound-fanout pattern.

Used neo4j-database integration

Used neo4j-database integration

So 192.168.3.104 is the focal node — every single conversation in the capture involves it. It either initiated or received ~16,400 separate conversations. Let me dig into the peers, ports, and traffic volumes.

Read Neo4j Cypher

Read Neo4j Cypher

One internal IP talking to **16,405 distinct peer addresses**. That alone is forensically extraordinary. Let me characterize the ports used.

Read Neo4j Cypher

Read Neo4j Cypher

Major finding: 192.168.3.104 is making **16,402 connections to port 445 (Microsoft-DS / SMB)** — the protocol exploited by WannaCry, NotPetya, EternalBlue, and many SMB-based worms. The ephemeral source ports (1573, 1577, 1578…) are sequential — typical of a single host opening rapid outbound sockets.

Let me confirm direction (104 initiating to 445, or external scanning 104).

Used neo4j-database integration

Used neo4j-database integration

Confirmed: 192.168.3.104 uses ephemeral source ports starting from ~1071 and rolling upward — typical client-side outbound sockets. The peer hosts each have port 445 — the **destination**. So 192.168.3.104 is **sweeping the 192.168.0.0/16 space, attempting SMB connections to ~16,400 distinct destination IPs**.

Let me check the bytes-sent/received distribution to characterize the conversations.

Used neo4j-database integration

Used neo4j-database integration

Very telling: **8,452 out of 8,473 outbound conversations send between 100–999 bytes**. This is the classic signature of a **scan/probe** (SMB SYN handshake + small protocol exchange), not real file transfer. But a handful (2 of >10k bytes, 11 of 1k–10k) are larger — likely **successful exploitations or actual payload transfers**. Let me look at those.

Read Neo4j Cypher

Read Neo4j Cypher

Striking detail: bytes_sent equals bytes_received in every single conversation — this means the field likely represents total bytes per flow, not directional. Still, the standouts are critical:

1. **200.10.122.10 — 1.16 MB transferred** to a public (Argentine) IP. This is the **only sizeable external connection** and a strong candidate for C2 / exfiltration / payload download.

2. **192.168.3.255 — broadcast address** (54 KB) — broadcast traffic, possibly NetBIOS name resolution.

3. **239.255.255.250 — SSDP multicast** (1.9 KB) — normal LAN discovery noise.

4. **72.5.123.29 and 66.235.128.158** — these are two of the "Host" nodes — small bytes.

5. Eight internal hosts received exactly 1152 bytes each — possibly the few peers that actually accepted the SMB handshake far enough to exchange a small payload.

Let me investigate 200.10.122.10 and the seven internal peers that returned more than minimal data.

Used neo4j-database integration

Used neo4j-database integration

Two of the public hosts (200.10.122.10, 200.69.128.1) are in Buenos Aires, Argentina; three are in Dallas. Let me check what ports the Dallas hosts connect on (they could be CDN/cloud providers, web services, or DNS).

Used neo4j-database integration

Used neo4j-database integration

Interesting — **66.252.13.214 (Dallas) sent 144 KB *inbound* to 192.168.3.104**, and 192.168.3.104 sent **1.16 MB** to 200.10.122.10 (Buenos Aires). Two notable external endpoints.

Let me also examine the 8 internal hosts that received 1152 bytes — that uniform payload size suggests successful protocol-level interaction (perhaps SMB session setup + tree connect responses).

Read Neo4j Cypher

Read Neo4j Cypher

Each of these 8 internal hosts has port **445** open and **4 additional ephemeral source ports** — meaning 192.168.3.104 connected to them **5 times each** at SMB (probably retry/exploit attempts). These 8 hosts are the **prime candidates for compromise — they answered**.

Let me check whether any of those 8 hosts themselves initiated outbound traffic (lateral movement / second-stage propagation).

Read Neo4j Cypher

Read Neo4j Cypher

No second-stage propagation in this capture window — the worm hasn't pivoted yet (or the capture ended before propagation).

Let me also look at the broadcast and multicast traffic, plus the 6-port and 7-port sets which seem to use "unknown" ports:

Used neo4j-database integration

Used neo4j-database integration

Outbound communication outside the LAN is *only* with these few external IPs — most volume going to 200.10.122.10.

Now let me check whether the local IP 192.168.3.104 itself is the only "infected" host or if there's any other anomaly:

Used neo4j-database integration

Used neo4j-database integration

These are normal RST/ACK responses — hosts replying once to a probe.
Confirms 192.168.3.104 is the sole originator of the suspicious activity. The few peers that "sent" 2 conversations did so only as half-replies to the original probes.

Now I have enough evidence. Let me cross-reference with MITRE ATT&CK. I know the relevant techniques from training, but should verify they're still listed under current technique IDs in the live ATT&CK matrix:

Searched the web

Searched the web

Excellent — I've confirmed the relevant ATT&CK techniques. The current matrix is v19. Let me also verify the SMB-exploitation and exfiltration technique IDs:

Searched the web

Searched the web

I have enough verified ATT&CK references. Let me also do one quick check on whether 192.168.3.104 has any normal-looking baseline traffic (DNS, HTTP, etc.) to characterize what kind of host it is.

Read Neo4j Cypher

Read Neo4j Cypher

This is the final piece. 192.168.3.104's entire activity is essentially one thing: **SMB port-445 contact attempts**, plus a single HTTP and single HTTPS connection (likely the C2 fetches to the Dallas hosts).

I now have a comprehensive forensic picture. Let me also generate a quick visual to support the verdict, then write up the analysis.

Verdict: strong evidence of an active malware infection on the local network

The capture shows a textbook **SMB-worm / lateral-movement campaign** in progress, almost certainly a host infected with malware in the EternalBlue / WannaCry / Conficker family.

What the graph reveals

A single internal host — 192.168.3.104 — is responsible for the entire capture. It is the sender or receiver in all 16,411 conversations, contacting 16,405 distinct peer IP addresses. No other internal host originates traffic; the few "replies" from other LAN hosts are 0-byte half-responses to its probes.

The traffic profile is overwhelmingly SMB scanning. 16,402 of the conversations target TCP **port 445 (Microsoft-DS)**. The host opens ephemeral source ports in sequential order (1071, 1072, 1073…1573, 1577, 1578…) — a classic signature of a tight client-side scanning loop. 99.7% of the outbound flows carry 100–999 bytes — far too small for legitimate file sharing, exactly the size of an SMB SYN + session-setup probe that gets aborted.

Eight internal hosts apparently responded to the SMB probe, exchanging exactly 1,152 bytes each across five separate connections per host: 192.168.217.181, 192.168.119.102, 192.168.118.121, 192.168.151.17, 192.168.71.158, 192.168.78.31, 192.168.159.221, 192.168.202.237. These are the **most likely candidates for compromise** — they completed enough of the SMB handshake for a payload-sized exchange. None of them initiated further traffic during the capture, so the worm has either not yet pivoted from them, or the capture ended too soon to see it.

An anomalous outbound transfer to Argentina: 192.168.3.104 sent **1.16 MB to 200.10.122.10** (host10.advance.com.ar, Buenos Aires) on an unrecorded port. This is the only large external flow in the capture and the strongest candidate for **command-and-control traffic, payload retrieval, or data exfiltration**. A smaller (144 KB) inbound flow from 66.252.13.214 (Dallas) toward 192.168.3.104 is also notable and may be a staging-server fetch.

Mapping to MITRE ATT&CK

Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation — the behavior of 192.168.3.104 matches **T1046 Network Service Discovery** (tactic TA0007 Discovery) almost perfectly. The scan pattern is the version commonly used by worms: specific ports (22, 21, 23, 135, 445, 1433, 1434, 3306, 3389, 5800, 5900, 5958, 5986) of several hosts — port 445 is one of the canonical malware-scan targets. MITREPtsecurity

The presence of large data exchanges (1152 bytes) with a small set of peers immediately after the scan corresponds to **T1210 Exploitation of Remote Services** (tactic TA0008 Lateral Movement). As MITRE notes, adversaries may exploit remote services to gain unauthorized access to internal systems once inside of a network… A common goal for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system. An adversary may need to determine if the remote system is in a vulnerable state, which may be done through Network Service Discovery. The SMB protocol is explicitly called out: there are several well-known vulnerabilities that exist in common services such as SMB, with MS17-010 / EternalBlue being the most famous example. MITREMITRE

The 1.16 MB outbound flow to Argentina aligns with **T1071 Application Layer Protocol** (TA0011 Command and Control) and/or **T1041 Exfiltration Over C2 Channel** (TA0010 Exfiltration).

A plausible mapped kill-chain for this host:

Stage	ATT&CK ID	Evidence in the graph
Discovery	T1046 — Network Service Discovery	16,402 outbound TCP 445 probes to distinct internal IPs from sequential ephemeral ports

Stage	ATT&CK ID	Evidence in the graph
Lateral movement	T1210 — Exploitation of Remote Services (SMB)	8 internal hosts each completed five 1,152-byte SMB exchanges
Command & Control	T1071 — Application Layer Protocol	Inbound 144 KB from 66.252.13.214 (Dallas); single HTTPS conversation to 66.235.128.158
Exfiltration / payload	T1041 — Exfiltration Over C2 Channel	1.16 MB to 200.10.122.10 (Buenos Aires)

Recommended response actions

1. **Isolate 192.168.3.104 immediately** — disconnect from the network, image the disk, and run AV/EDR triage; this host is almost certainly infected.

2. **Quarantine and inspect the eight responding hosts** (192.168.217.181, 192.168.119.102, 192.168.118.121, 192.168.151.17, 192.168.71.158, 192.168.78.31, 192.168.159.221, 192.168.202.237) — they are the most likely second-stage victims.

3. **Block egress** to 200.10.122.10 (Buenos Aires) and 66.252.13.214 (Dallas) at the perimeter firewall and capture any further DNS/HTTP requests for forensic correlation.

4. **Patch SMB / disable SMBv1** across the LAN (MS17-010 / KB4013389) and require SMB signing.

5. **Search for persistence artifacts** on 192.168.3.104 (scheduled tasks, services, WMI subscriptions, registry Run keys) — discovery-only behavior in this capture doesn't rule out an already-established foothold.

Bottom line: this is not a benign network. The capture shows a single host actively scanning ~16,400 SMB targets across the LAN, succeeding against at least eight, and exchanging more than a megabyte of data with an external host in Argentina. The behavioral fingerprint is highly consistent with an SMB-propagating worm or post-exploitation lateral-movement tool, and warrants treatment as an active incident.

MC-90-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

12:46 AM

Claude responded: I'll analyze the network traffic captured in the Neo4j database against the MITRE ATT&CK framework to identify potential malicious activity.

I'll analyze the network traffic captured in the Neo4j database against the MITRE ATT&CK framework to identify potential malicious activity. Let me start by exploring the database schema.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

Good — I have the schema. Let me explore the network structure: hosts, domains, IPs (especially flagged ones), ports, and conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — those don't look like local hosts; they look like external endpoints. Let me check what's actually local.

Used neo4j-database integration

Used neo4j-database integration

Hmm — the dataset has 16,401 "local" IPs and only 5 external ones. That's the opposite of what I'd expect for a typical LAN capture. Let me re-examine: in many flow-capture datasets, the is_Local field actually flags the *organization's* IPs (which can include broad public ranges if it's a hosting provider). Let me look more carefully.

Used neo4j-database integration

Used neo4j-database integration

So the local network is a big 192.168.0.0/16 with thousands of internal IPs. The 5 external IPs are surprisingly few — this means most of the traffic is internal. Let me now look at port usage and conversation patterns.

Read Neo4j Cypher

Read Neo4j Cypher

Very significant finding. Essentially every conversation in the capture (16,402 of 16,411) targets **TCP port 445 (Microsoft-DS / SMB)**. That is an enormous red flag — SMB is the classic vector for several wormable exploits (EternalBlue / MS17-010, used by WannaCry, NotPetya, EmotetSMB, etc.). Let me characterize the SMB traffic more precisely.

Read Neo4j Cypher

Read Neo4j Cypher

This is the smoking gun. A single host, **192.168.3.104**, initiated **16,402 SMB conversations** with **8,466 distinct internal targets**. That is textbook SMB sweeping/horizontal scanning behaviour. Let me confirm and look at the data shape.

Read Neo4j Cypher

Read Neo4j Cypher

I have more than enough now. Let me check a few more characteristics to round out the picture (response size on the SMB sweep, the few non-internal flows, and whether 192.168.3.104 ever received traffic), then build the report.

Used neo4j-database integration

Used neo4j-database integration

Critical confirmation: 95% of those 16,402 SMB conversations are ≤576 bytes, with a median of just 288 bytes. **These are not file transfers — they're tiny connection-attempt-sized packets.** That's exactly what an SMB worm or scanner produces while probing for vulnerable hosts. Let me check a few final pieces.

Read Neo4j Cypher

Read Neo4j Cypher

Very interesting — there are external flows tied to host 192.168.3.104:

- **66.252.13.214 → 192.168.3.104**: ~145 KB inbound. This looks like a payload/file delivered to the host (possible initial infection or downloaded loader).
- **192.168.3.104 → 200.10.122.10**: ~1.16 MB outbound. Large exfil-shaped flow.
- **192.168.3.104 → 72.5.123.29, 66.235.128.158, 200.69.128.1**: small flows consistent with C2 beaconing / DNS-over-HTTP / web check-ins.

Let me look at what ports these external flows used.

Read Neo4j Cypher

Read Neo4j Cypher

This is a striking pattern. The host **192.168.3.104** has:

1. An inbound transfer of **~145 KB from external IP 66.252.13.214** on the high port pair (1027, 1333, 1336, 1573, 2081 ↔ 1071) — that size and high-port distribution is highly characteristic of a malware payload being delivered to the host.
2. A small HTTP (port 80) call out to **72.5.123.29** — beacon/check-in pattern.

3. A small HTTPS (port 443) call out to **66.235.128.158** — beacon/check-in pattern.

4. A large **~1.16 MB outbound** transfer to **200.10.122.10** with no recognised standard port — classic data exfiltration shape.

5. A small UDP-style flow to **200.69.128.1** — possibly a name-resolution or status check.

Combined with the **8,466-host SMB sweep on TCP/445**, this is a complete textbook attack chain. Now let me check whether multiple infected hosts exist and confirm no other host shows the same multi-stage behaviour.

Read Neo4j Cypher

Read Neo4j Cypher

Confirmed: **only one internal host** is doing the SMB sweeping. Now I have everything I need. Let me also verify briefly against MITRE ATT&CK by searching for the relevant techniques.

Searched the web

Searched the web

I have everything I need to deliver a confident verdict. The MITRE references confirm the relevant techniques. Let me write the report.

Verdict: Yes — strong indications of malware/cyber-attack activity in the captured network

The graph shows behaviour that is not just suspicious — it is, with very high confidence, an active **SMB-worm / lateral-movement attack** in progress, originating from a single compromised internal host, with evidence of an external command-and-control / staging relationship and possible data exfiltration on the side.

The compromised host

192.168.3.104 is the source of essentially the entire attack pattern. It is the **only** internal host of the 16,401 captured that initiates SMB conversations to more than a handful of peers, while it alone is responsible for **16,402 of 16,411 conversations** in the dataset (≈99.95%).

What it is doing — and where each step maps to MITRE ATT&CK

1. Internal network sweeping on TCP/445 (SMB)

- 16,402 conversations from 192.168.3.104 to TCP/445.

- **8,466 distinct destination IPs** inside 192.168.0.0/16.

- Per-conversation payload is tiny: median 288 bytes, p95 576 bytes, max 1,152 bytes — i.e. session-setup / negotiate-protocol-sized packets, *not* file transfers.

That signature — one host fanning out to thousands of internal peers on port 445 with minimal payloads — is the textbook fingerprint of an SMB scanner / worm propagation engine. It maps to multiple ATT&CK techniques simultaneously:

- **T1046 — Network Service Discovery** (Tactic: *Discovery*): the host is enumerating which internal systems answer on SMB.

- **T1021.002 — Remote Services: SMB / Windows Admin Shares** (Tactic: *Lateral Movement*): SMB on TCP/445 targets `ADMIN$`, `C$`, `IPC$` $shares. This technique falls under the Lateral Movement tactic, which means attackers use it to move$ "$6ee2954a-b384-44c4-a31d-93312bcdd5de$" $injected =$ "$space$", $and is heavily abused for worm-style spread—Conti spreads via SMB to encrypt files on multiple hosts simultaneously … Emotet has$, C$, and IPC$ shares for lateral movement :antCitation[]{citations="df94c5c3-3c68-4a54-861f-dd6605a5b7bf" injected="space"}.

- **T1210 — Exploitation of Remote Services** is also a strong candidate if the SMB connections are not authenticating but exploiting (EternalBlue / MS17-010 family).

The fact that the sweep reaches a near-uniform 73–82 hosts per /24 subnet across the entire 192.168.x.x range is consistent with a worm walking the address space rather than an authenticated administrator with a target list.

2. External command-and-control / staging behaviour

Of the 5 non-local IPs in the capture, **all 5 communicate with 192.168.3.104** (and only with it):

External IP	Direction	Volume	Ports	Likely role (ATT&CK)
66.252.13.214	**inbound to .104**	~145 KB	high ports (1027, 1071, 1333, 1336, 1573, 2081)	Payload / tooling delivery — **T1105 Ingress Tool Transfer**
72.5.123.29	outbound from .104	~3.5 KB	TCP/80 (HTTP)	C2 beacon — **T1071.001 Web Protocols**
66.235.128.158	outbound from .104	~2.8 KB	TCP/443 (HTTPS)	C2 beacon (encrypted) — **T1071.001 / T1573 Encrypted Channel**
200.69.128.1	outbound from .104	~0.7 KB	unspecified	likely C2 check-in / status

External IP	Direction	Volume	Ports	Likely role (ATT&CK)
200.10.122.10	**outbound from .104**	**~1.16 MB**	unspecified	**T1041 Exfiltration over C2 channel** — by far the largest single transfer in the capture

The 1.16 MB outbound to 200.10.122.10 is suspicious in its own right: it dwarfs every other flow, and it goes to an external address that is never contacted by any other internal host.

3. Putting the chain together (MITRE ATT&CK lifecycle)

The graph is consistent with the following attack lifecycle on 192.168.3.104:

1. **Initial Access / Tool Delivery (T1105)** — ~145 KB pulled from 66.252.13.214 over high ports.

2. **Command and Control (T1071.001, T1573)** — HTTP and HTTPS check-ins to two distinct external hosts; one additional low-volume callback.

3. **Discovery (T1046)** — the host enumerates the rest of the 192.168.0.0/16 space on SMB.

4. **Lateral Movement (T1021.002 / T1210)** — it issues SMB connections to 8,466 distinct internal peers, the textbook propagation step used by Conti, Emotet, WannaCry, NotPetya, and similar SMB-borne malware.

5. **Exfiltration (T1041)** — a single ~1.16 MB outbound transfer to a fourth external IP.

Caveats and what the graph cannot tell us

A few honest limitations worth flagging:

- The capture only contains *flow-level* metadata (sender, receiver, ports, byte counts) — there is no payload, no SMB dialect/command, no timestamps in what was queried, and no authentication outcome. So I cannot prove whether the SMB connections succeeded, failed, or exploited a vulnerability — only that the *fan-out pattern* is highly anomalous and matches malware behaviour better than any legitimate administrative use I can think of.

- The dataset's is_known_attacker, is_spam, and is_Bot flags are all false for every IP, including the external ones. Those reputation flags appear to be unpopulated rather than affirmatively clean — they should not be read as exonerating evidence.

- One host alone showing this pattern is consistent with the *first stage* of a worm — the rest of the network may not yet be compromised, or the capture window may simply end before propagation succeeded. Either way, the originator (192.168.3.104) should be treated as compromised.

Recommended immediate actions

1. **Isolate 192.168.3.104** from the network — yank the cable / quarantine via NAC. Do not power it off until forensic memory capture is considered.

2. **Block egress** to 66.252.13.214, 72.5.123.29, 66.235.128.158, 200.69.128.1, and especially 200.10.122.10 at the perimeter, and add them to the threat-intel watchlist.

3. **Hunt the rest of the LAN** for connections *from* 192.168.3.104 on port 445 that resulted in subsequent outbound activity from the *target* host — those would be the second-stage infections.

4. **Patch / disable SMBv1** and apply MS17-010 if not already; verify that the SMB sweep was being blocked at the host level on the 8,466 contacted machines.

5. **Acquire disk + memory image** from 192.168.3.104 for malware identification before remediation.

Bottom line: the network was, during the capture window, hosting an active SMB-based malware propagation event from 192.168.3.104, almost certainly accompanied by external C2 and a probable data-exfiltration event. This is not a benign anomaly — it warrants incident response.

MC-90-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

1:00 AM

Claude responded: I'll analyze the network traffic data in the Neo4j database and cross-reference it with MITRE ATT&CK techniques to detect potential cyber-attacks.

I'll analyze the network traffic data in the Neo4j database and cross-reference it with MITRE ATT&CK techniques to detect potential cyber-attacks. Let me start by loading the database tools and exploring the schema.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

Good. Let me explore the data to understand what we're working with.

Used neo4j-database integration

Used neo4j-database integration

Interesting — 16,401 local IPs and only 5 external IPs (which match the 5 hosts). Let me look at the local IPs:

Used neo4j-database integration

Used neo4j-database integration

Very interesting. The "local" network has 16,401 IPs across the **entire 192.168.0.0/16 space** — that's not a normal LAN, that's likely the result of randomized/scanning behavior. Also note 224.0.0.x and 239.255.255.x (multicast). Let me look more carefully at the data structure and what counts as local vs external.

Read Neo4j Cypher

Read Neo4j Cypher

Hmm — only 5 external IPs are marked, but the hosts include host10.advance.com.ar (200.10.122.10). Wait — that hostname implies the LOCAL machine is actually 200.10.122.10 (a public IP in Argentina). The "is_Local" flag here may actually be inverted from what one would expect, or the labeling refers to something else. Let me check carefully:

Read Neo4j Cypher

Read Neo4j Cypher

OK so the 5 external IPs match the 5 hosts. Let me look at who's communicating with whom:

Used neo4j-database integration

Used neo4j-database integration

This is a strong signal already — many different "senders" all going to the same 192.168.3.104 with identical 80-byte payloads. Let me dig in.

Used neo4j-database integration

Used neo4j-database integration

Massive red flag. 192.168.3.104 is both sending and receiving thousands of conversations, while every other "IP" sends/receives exactly 1. That's the textbook signature of a host either scanning the network or being scanned. Let me characterize it further:

Used neo4j-database integration

Used neo4j-database integration

192.168.3.104 uses 3,976 distinct ports. All other hosts use exactly 1 port each. Let's see the port range:

Read Neo4j Cypher

Read Neo4j Cypher

Ports 1026–5000 — this is the classic Windows ephemeral port range (Windows XP/2003 used 1025–5000 for dynamic outbound ports). Let me look at what other IPs talk to:

Read Neo4j Cypher

Read Neo4j Cypher

Smoking gun. Every single one of the 16,400 "other" IPs in the 192.168.0.0/16 space is being contacted on **port 445 (SMB / Microsoft-DS)** — and exactly once each. Combined with the source 192.168.3.104 using thousands of sequential ephemeral source ports, this is unmistakably an **SMB port scan / SMB-worm propagation pattern** sweeping the entire private address space.

Let me confirm the directionality and packet sizes:

Used neo4j-database integration

Used neo4j-database integration

The 80-byte and 288-byte packets dominate. 80 bytes = TCP SYN packets typically (3-packet handshake ≈ 80 bytes one direction at L2 with no payload). 288 bytes = a small SMB negotiate-protocol probe. Let me check whether 192.168.3.104 talks to the external hosts too:

Read Neo4j Cypher

Read Neo4j Cypher

Very telling. Notably:

- 192.168.3.104 sends a heavyweight **1.16 MB** to 200.10.122.10 (host10.advance.com.ar in Argentina) — that's likely **data exfiltration** or a malware download.

- 66.252.13.214 (US, Dallas) pushes **144 KB** down to 192.168.3.104.

Let me look at the ports these external conversations used:

Read Neo4j Cypher

Read Neo4j Cypher

Very interesting:

- 72.5.123.29: HTTP (port 80) — possible C2 or download
- 66.235.128.158: HTTPS (port 443) — possible C2 over TLS
- 66.252.13.214: ports in the ephemeral range — possible peer-to-peer / C2 callback
- 200.10.122.10 and 200.69.128.1: null port — possibly ICMP or non-TCP

Now let me check what other types of traffic exist (any non-445 outgoing port traffic in the local space):

Read Neo4j Cypher

Read Neo4j Cypher

16,402 conversations on SMB (port 445) — that's effectively all the local traffic. The remaining ~16,400 conversations between the external hosts and 192.168.3.104 use scattered ephemeral ports plus HTTP/HTTPS.

Let me now query the MITRE ATT&CK matrix for relevant techniques:

Searched the web

Searched the web

Let me also look up the specific MITRE techniques most relevant to what we're observing:

Searched the web

Searched the web

Excellent — I have everything I need. Let me also quickly do one or two more sanity checks on the data, especially around the external host connections and whether any of those IPs are flagged as malicious:

Used neo4j-database integration

Used neo4j-database integration

Decisive finding: 100% of all 16,411 conversations in the capture involve 192.168.3.104. This is not a generic network capture — it is a capture centered on a single endpoint, and that endpoint is doing essentially nothing but scanning.

Now let me check the multicast/broadcast addresses to make sure I'm not missing anything:

Read Neo4j Cypher

Read Neo4j Cypher

That's SSDP multicast (UPnP discovery) — also from 192.168.3.104. Consistent with discovery activity.

I now have a complete and very clear picture. Let me present the verdict.

Verdict: Strong indications of a cyber-attack are present.

The host **192.168.3.104** is unambiguously the source of malicious activity, and it is almost certainly **infected with malware exhibiting worm-like SMB-propagation behavior**.

What the graph shows

The capture is centered on a single host. All 16,411 CONVERSATION nodes in the graph touch 192.168.3.104 as sender or receiver. The "16,401 local IPs" in the graph are not really endpoints in a LAN — they are 16,401 distinct scan *targets* across the entire 192.168.0.0/16 range, each appearing in exactly one conversation, each receiving an 80-byte probe.

192.168.3.104 exhibits a near-textbook scanning fingerprint:

Indicator	**Value**	**What it means**
Outbound conversations	8,473	Massive fan-out
Unique destinations contacted	8,473	One probe per host — a sweep, not legitimate traffic
Distinct source ports used	3,976, spanning **1026–5000**	Walking the Windows ephemeral port range — typical of automated/scripted scanning
Destination port on **every** non-3.104 IP	**445/tcp (SMB)** — 16,396 of them	Targeting SMB exclusively
Typical probe size	80 bytes (SYN) and 288 bytes (SMB negotiate)	Half-open probes + SMB protocol negotiation
Multicast traffic to 239.255.255.250	Present	SSDP / UPnP discovery, also from 3.104

No other internal IP initiates more than one conversation; they only ever appear as scan targets answering 80-byte replies (SYN-ACK / RST). The traffic pattern is one-to-many, not many-to-many — i.e. exactly what an infected host doing reconnaissance looks like, not normal user traffic.

Mapping to MITRE ATT&CK

The graph maps cleanly to several MITRE techniques, in this order of confidence:

- **T1046 – Network Service Discovery (Discovery tactic).** Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port, vulnerability, and/or wordlist scans. Detection guidance for T1046 explicitly calls out "high volume of SYN packets without corresponding ACK packets … multiple failed connection attempts to sequential or random ports … repeated connection attempts originating from a single host to multiple destinations" — all three are present here. The exclusive choice of port 445 is also notable: "Service discovery identifies the specific authentication and management protocols accessible on each host: SMB on port 445, RDP on 3389, SSH on 22, WinRM on 5985, VNC on 5900. Each of these is a potential lateral movement path." MITRE + 2

- **T1021.002 – Lateral Movement: SMB / Windows Admin Shares.** SOC detection guidance for this technique reads almost like a description of 192.168.3.104's behavior: "Monitor for rapid sequential SMB connections from a single source to multiple destinations, which indicates automated lateral movement tools like CrackMapExec, PsExec, or worm-like spreading behavior from Emotet, Conti, or NotPetya." Startupdefense

- **T1071.001 – Application Layer Protocol: Web Protocols (C2).** Outbound conversations from 192.168.3.104 to the external hosts use HTTP (port 80 → 72.5.123.29) and HTTPS (port 443 → 66.235.128.158) in addition to obscure ephemeral ports — a typical C2/check-in pattern.

- **T1041 – Exfiltration Over C2 Channel** (possible). 192.168.3.104 sends **1,163,472 bytes (~1.1 MB)** outbound to 200.10.122.10 (host10.advance.com.ar, Argentina) — orders of magnitude larger than any other flow in the capture and the only large outbound payload. Either malware download pulled in or, more concerningly, data being shipped out.

Why this is almost certainly malware, not a benign admin

A legitimate vulnerability scanner does not:

- run from a non-server endpoint in the user subnet,

- sweep the entire 192.168.0.0/16 private space (most of which is unassigned to your organization),
- combine that sweep with HTTP/HTTPS check-ins to **five different external IPs** in Dallas and Buenos Aires,
- and ship 1+ MB to one of those external hosts.

That combination — internal SMB worming + external C2 callouts + a single large outbound transfer — is the canonical kill-chain shape of an SMB-propagating worm/bot (the EternalBlue/Conficker/WannaCry/NotPetya family). The use of ports 1026–5000 as source ports specifically points to a Windows XP / Server 2003 era host, which is consistent with the historical SMBv1 exploitation timeframe.

Recommended actions

1. **Isolate 192.168.3.104 immediately** — disconnect from the network, do not power off (preserve volatile memory for forensics).
2. **Block egress to all five external IPs**: 66.252.13.214, 72.5.123.29, 200.10.122.10, 66.235.128.158, 200.69.128.1 — and treat them as C2 IOCs to be shared with your IR team / threat-intel feed.
3. **Block intra-subnet 445/tcp at the network edge / VLAN ACL** until the scope of infection is confirmed; check whether any host *responded* to the SMB probes (the 80-byte replies suggest at least some did at the TCP level — investigate whether SMB sessions were actually established).
4. **Forensic the host:** memory image, examine running services for unexpected SMB-using processes (Conficker-style scheduled tasks, PsExec-style services, EternalBlue-related drivers), pull the autoruns and recent file-write history. Pay special attention to whatever generated the 1.1 MB outbound payload to 200.10.122.10.
5. **Patch SMB/EternalBlue** (MS17-010) and disable SMBv1 across the environment if not already done.
6. **Threat-hunt the rest of the network** for the same outbound IPs and for any other host showing one-to-many port-445 fan-out.

Bottom line: this is not benign traffic. The capture shows a single compromised endpoint actively scanning the local network over SMB while maintaining external command-and-control and exfiltrating data — a high-confidence malware infection requiring immediate incident response.

MC-90-ns2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

1:24 AM

Claude responded: I'll analyze the Neo4j database to look for indicators of malicious activity in the network traffic.

I'll analyze the Neo4j database to look for indicators of malicious activity in the network traffic. Let me start by exploring the schema and understanding what data we're working with.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

Good, I have the schema. Now let me understand the network structure: Hosts, OS, Domains, IPs (local & remote), Ports, and Conversations between them. Let me explore the data.

Read Neo4j Cypher

Read Neo4j Cypher

Hmm, these all look like cloud hosts, not local computers. Let me check the local network.

Used neo4j-database integration

Used neo4j-database integration

Interesting - the schema labels are misleading. The "Hosts" are actually external/cloud endpoints. The 16401 "local" IPs are the local network. Let me investigate further.

Used neo4j-database integration

Used neo4j-database integration

Wait — 16,401 distinct 192.168.x.y addresses across ~256 subnets is unusual. A real LAN normally uses one or two /24s. This is suspicious by itself. Let me look more closely at the conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Note: every conversation has equal bytes_sent and bytes_received. That's odd. Let me see actual sender/receiver patterns.

Read Neo4j Cypher

Read Neo4j Cypher

A very strong pattern: a single host **192.168.3.104** is receiving traffic from huge numbers of distinct IPs, all sending exactly 80 bytes. Let me quantify.

Read Neo4j Cypher

Read Neo4j Cypher

A massive smoking gun: **192.168.3.104 received 7,938 inbound conversations from 7,938 distinct unique source IPs**. Now let me look at the reverse — who is the biggest sender?

Read Neo4j Cypher

Read Neo4j Cypher

Now the picture is crystal clear:

- **192.168.3.104** sent **8,473** conversations to **8,473 unique receivers** (288 bytes each)
- **192.168.3.104** received **7,938** conversations from **7,938 unique senders** (80 bytes each)
- Every other IP in the network participated in just **one** conversation total

Let me check the ports involved.

Read Neo4j Cypher

Read Neo4j Cypher

Port 445/SMB — Microsoft-DS — is the destination for nearly every conversation. This is the exact port abused by SMB worms (e.g., Conficker, WannaCry/EternalBlue). Let me confirm the topology around 192.168.3.104.

Used neo4j-database integration

Used neo4j-database integration

Now the picture is essentially complete. Let me check a couple more things — bytes patterns and external (non-local) communication.

Used neo4j-database integration

Used neo4j-database integration

More indicators. The infected host 192.168.3.104 also communicated with **5 external IPs**, and **1.16 MB was uploaded to 200.10.122.10** — a sizeable exfiltration or C2 channel. Let me check what the external hosts are.

Used neo4j-database integration

Used neo4j-database integration

The big exfiltration target (1.16 MB) is host10.advance.com.ar (200.10.122.10). Now let me cross-reference with the MITRE ATT&CK framework.

Searched the web

Searched the web

Excellent — the references confirm that Network Service Discovery (T1046) involves scanning to identify services on remote hosts, with Conficker specifically cited as a worm using this technique. Let me check one more thing — whether 192.168.3.104 also sources outbound traffic from a wide range of source ports (consistent with a scan). MITRE

Used neo4j-database integration

Used neo4j-database integration

That clinches it. No other host on the LAN sent traffic to more than one destination. The entire scanning activity originates from a single compromised host. Let me finalise a couple of figures for the verdict.

Read Neo4j Cypher

Read Neo4j Cypher

I have everything I need. Let me also produce a visualization of the attack pattern.

V

visualize

V

visualize show_widget

Verdict: The capture shows strong indicators of an active malware infection on 192.168.3.104 — almost certainly an SMB-propagating worm.

The pattern in the graph is unambiguous. Out of 16,411 conversations, **every single one** involves the same host, 192.168.3.104, and **99.95% of them target TCP port 445 (SMB / Microsoft-DS)**. No other LAN host in the capture talked to more than one destination.

What the graph actually shows

A single host scanning the entire local /16. 192.168.3.104 initiated 8,473 outbound conversations to 8,473 distinct destinations spread across 262 different /24 subnets (192.168.0.0/16). Each outbound connection sent the same 288-byte payload to port 445. That is not normal client behaviour — even an aggressive enterprise vulnerability scanner does not look like this against a live LAN, and a legitimate user workstation never does. The uniform packet size, the single destination port, and the saturation of the address space are the signature of an automated SMB sweep.

Responses from 7,938 of those targets. Of the IPs scanned, 7,938 sent back an 80-byte reply — i.e. a TCP-level response indicating a host was reachable. The 535-host gap between probes sent and responses received is consistent with empty IP space or filtered hosts. This is exactly what a worm's target list looks like *after* the scan: hosts that answered are now candidates for exploitation.

Outbound contact with five external IPs. 192.168.3.104 was the only LAN host talking to the public Internet. The volumes are revealing:

- **1.16 MB uploaded** to host10.advance.com.ar (200.10.122.10) — strongly suggestive of data exfiltration or upload to a drop site.

- **144 KB inbound** from 66.252.13.214 across six different ephemeral ports — the right shape for a C2 channel or staged-payload download.
- Three smaller outbound conversations to 72.5.123.29, 66.235.128.158, and 200.69.128.1.

Mapping to MITRE ATT&CK

Tactic	Technique	Evidence in the graph
Discovery	**T1046 — Network Service Discovery**	Sweeping 8,473 IPs across 262 subnets on a single port to enumerate SMB-reachable hosts. MITRE explicitly lists Conficker as a worm that uses T1046 MITRE.
Discovery	**T1018 — Remote System Discovery**	Identifying which IPs on the LAN are alive (the 7,938 responders).
Lateral Movement	**T1021.002 — SMB / Windows Admin Shares**	Port 445 is the only service touched at scale. T1021.002 describes adversaries establishing authenticated SMB sessions on port 445, a technique used by worms such as Emotet and HermeticWizard to spread Startupdefense.
Lateral Movement	**T1210 — Exploitation of Remote Services**	The combination of mass port-445 contact and small, fixed-size payloads is the behaviour of automated SMB exploit modules (the family includes EternalBlue, Conficker's MS08-067 vector, etc.). NotPetya famously combined SMB-based propagation with EternalBlue to spread laterally Elisity.
Command and Control	**T1071 / T1571**	Sustained bidirectional traffic with 66.252.13.214 over multiple non-standard ports fits a C2 callback.
Exfiltration	**T1041 — Exfiltration Over C2 Channel** (or **T1567** over web service)	1.16 MB outbound to a single external host (200.10.122.10) — orders of magnitude more than any other external conversation.

Caveats

A few features of the capture should be flagged: every conversation has bytes_sent == bytes_received, the LAN spans an unrealistically large 192.168.0.0/16 with 16,401 unique addresses, and only 5 conversations have repeating bytes_sent values that suggest the data was synthesised or summarised in some way. This means the capture may be from a lab/dataset rather than a production network, and the absolute byte counts shouldn't be over-interpreted. However, the *structural* finding — one host fanning out to thousands of others on port 445 while everyone else is silent — is real and can only be produced by an SMB scanner/worm.

Recommended response

192.168.3.104 should be treated as compromised: isolate it from the network, capture volatile memory and disk images for forensics, and check every host that replied on port 445 for signs of secondary infection (new services, scheduled tasks, admin-share writes, PSEXESVC artefacts). The external IPs (especially 200.10.122.10 / host10.advance.com.ar and 66.252.13.214) should be added to blocklists and submitted to threat-intelligence platforms for reputation lookup.

MC-90-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

1:34 AM

Claude responded: That's surprising — 16,401 "local" IPs and only 5 external.

That's surprising — 16,401 "local" IPs and only 5 external. Let me check this more carefully:

Used neo4j-database integration

Used neo4j-database integration

16,401 local IPs in 192.168.0.0/16 is far more than a typical LAN — likely a /16 supernet capturing many subnets, or these are connection endpoints. Let me look at top talkers and subnet distribution:

Read Neo4j Cypher

Read Neo4j Cypher

This response didn't load. Try again by chatting to Claude.

You said: The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time.

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

1:35 AM

Claude responded: Local IPs span the entire 192.

Local IPs span the entire 192.168.0.0/16 fairly evenly. That's already suspicious — a real LAN doesn't typically have ~16,000 hosts spread evenly across 256 subnets. Let me examine traffic patterns:

Read Neo4j Cypher

Read Neo4j Cypher

This is a smoking gun. A single host 192.168.3.104 is involved in 16,411 conversations — essentially *every conversation in the capture* — while every other internal IP only appears 1–2 times. Let me dig deeper:

Used neo4j-database integration

Used neo4j-database integration

3,976 distinct source ports — classic ephemeral-port behavior of a host opening thousands of outbound connections. Now let me look at *what ports* the OTHER side is using (i.e., the destination services this host hits):

Read Neo4j Cypher

Read Neo4j Cypher

Let me restructure — first look at what conversations look like, what ports are most used overall, and most importantly the destination IPs 192.168.3.104 reaches:

Used neo4j-database integration

Used neo4j-database integration

This is extraordinary. 192.168.3.104 is a hub talking to **every single other internal IP** — 8,473 distinct outbound destinations, 7,938 distinct inbound sources, **each peer touched exactly once or twice**. Let me check the byte sizes to characterize the activity:

Used neo4j-database integration

Used neo4j-database integration

Strong pattern: outbound is ~288 bytes (multiples of 96, typical of probe packets) and inbound is ~80 bytes (typical of small responses or RSTs). **And there's one anomalous 144,816-byte conversation** — let me find it:

Read Neo4j Cypher

Read Neo4j Cypher

An **external IP** (66.252.13.214 — one of the 5 non-local IPs, listed as a Host) sent 144 KB to 192.168.3.104. Let me check what port:

Read Neo4j Cypher

Read Neo4j Cypher

Now let me check the external IPs and what they're doing:

Used neo4j-database integration

Used neo4j-database integration

ALL external traffic involves 192.168.3.104 exclusively. No other internal host talks to the outside world. Let me see the external IP locations and ports:

Used neo4j-database integration

Used neo4j-database integration

Now I have geo context: external IPs are in Dallas, US and Buenos Aires, Argentina (advance.com.ar is an Argentine ISP). The 1.16 MB outbound transfer goes to Buenos Aires. Let me verify a couple of remaining patterns. First, the small "extra" conversations from other internal IPs — they may be part of the same scan but worth confirming:

Read Neo4j Cypher

Read Neo4j Cypher

Zero conversations don't involve 192.168.3.104. Every single conversation in the entire capture has 192.168.3.104 as either sender or receiver. This is overwhelming evidence of a single host scanning the entire network. Let me confirm port-scan behavior — what destination ports does it hit?

Used neo4j-database integration

Used neo4j-database integration

Confirmed. Every other internal IP (16,396 of them) uses port **445 — Microsoft-DS / SMB**. The host 192.168.3.104 is scanning every IP in 192.168.0.0/16 on port 445. This is the signature of a worm / lateral-movement scanner.

Let me consult the MITRE ATT&CK matrix to map this behavior to specific techniques:

Searched the web

Searched the web

Now let me also pull T1046 (Network Service Discovery) and T1048 (Exfiltration) to confirm the mapping:

Searched the web

Searched the web

I have enough to issue a confident verdict. Let me also confirm the small "extra" inbound conversations (1-2 conversations from many local IPs are responses) and quickly verify the suspected ransomware port pattern (the EternalBlue / WannaCry / NotPetya signature is *exactly* SMB-445 scanning of an entire /16):

Read Neo4j Cypher

Read Neo4j Cypher

Good — 192.168.3.104 itself uses 3,976 distinct ephemeral source ports (1027-5000-ish range, classic Windows ephemeral range). Combined with the fact that *all other* internal hosts use port 445, this is unambiguous: a Windows host firing ephemeral connections at port 445 of every IP in the /16.

Verdict: 🚨 Active SMB-based worm / lateral-movement attack in progress

The traffic capture shows behavior that is **inconsistent with any legitimate use of the network** and matches multiple MITRE ATT&CK techniques used by SMB-worm malware (WannaCry, NotPetya, Emotet, Conti, EternalBlue-class implants).

The smoking-gun pattern

Observation	Value
Internal host 192.168.3.104 involvement	**In 16,411 of 16,411 conversations (100%)**
Distinct internal peers it contacted (outbound)	8,473
Distinct internal peers contacting it (inbound)	7,938
Subnet coverage	The entire 192.168.0.0/16 — all 256 /24s, ~80 hosts per /24
Destination service on every other internal IP	**TCP/445 (Microsoft-DS / SMB)** — 16,396 hosts
Source ports used by 192.168.3.104	3,976 distinct ephemeral ports
Per-conversation payloads	288 bytes out / 80 bytes in (tiny probe + tiny response/RST)
External communications	Only 192.168.3.104 talks outside; **no other internal host**

A normal workstation does not open one SMB connection to ~8,500 unique hosts in a single capture. Legitimate file-server / domain-controller behavior shows the *opposite* fan-in: many clients talking to a few servers, not one host fanning out to everything.

Mapping to MITRE ATT&CK

TA0007 – Discovery → T1046 Network Service Discovery Sequential, fan-out probes from one source to thousands of destinations on a single well-known service port (445) with tiny payloads is the textbook signature MITRE describes — "rapid sequential connection attempts to multiple hosts on common service ports."

TA0008 – Lateral Movement → T1021.002 Remote Services: SMB / Windows Admin Shares Port 445 is the SMB / Admin-Shares channel. MITRE's guidance: "rapid sequential SMB connections from a single source to multiple destinations … indicates automated lateral movement tools like CrackMapExec, PsExec, or worm-like spreading behavior from Emotet, Conti, or NotPetya." This matches 192.168.3.104 exactly.

TA0008 – Lateral Movement → T1210 Exploitation of Remote Services (very likely) SMB/445 scans typically precede or accompany EternalBlue (MS17-010) or similar SMB exploits used by WannaCry, NotPetya, TrickBot, Emotet, and many ransomware families.

TA0011 – Command & Control → T1071 Application Layer Protocol *and* **TA0010 – Exfiltration → T1041 Exfiltration Over C2 Channel** (suspected) 192.168.3.104 is the **only** host with external connectivity, and it talks to five Internet endpoints in two countries — Dallas (US) and Buenos Aires (Argentina, ISP advance.com.ar). Two flows are particularly suspicious:

- 66.252.13.214:2081 (Dallas) → 192.168.3.104: **144,816 bytes inbound** on a non-standard high port. Consistent with a **payload / second-stage download or C2 push**.

- 192.168.3.104 → 200.10.122.10 (Buenos Aires): **1,163,472 bytes outbound (~1.16 MB)**. Consistent with **data exfiltration**.

The other two external flows (port 80 HTTP, port 443 HTTPS) are small and could be benign, but in this context they are likely beaconing.

Why this almost certainly is malware, not pen-testing or admin activity

1. **Coverage is exhaustive**, not targeted: every /24 in the entire 192.168.0.0/16 is hit. Administrators don't scan their whole network this way; they have inventories.

2. **One uniform port** (445) is hit on every peer. Vulnerability scanners produce many ports per host; this is **single-service fan-out**, the signature of an SMB-spreading worm.

3. **The scanner is also the only host with internet access**, and it both *receives* a sizeable inbound chunk from outside and *sends* an order-of-magnitude larger chunk outbound — exactly the C2-download-then-exfiltrate pattern.

4. **External endpoints are geographically dispersed** (US + Argentina) and the largest flow targets a residential/business ISP host in Argentina (host10.advance.com.ar), which is a very atypical destination for legitimate enterprise software updates.

Recommended response

1. **Immediately isolate 192.168.3.104** at the switch/firewall level.

2. **Block** the five external IPs at the perimeter: 66.252.13.214, 72.5.123.29, 200.10.122.10, 66.235.128.158, 200.69.128.1.

3. **Forensically image** 192.168.3.104 before powering off — examine running processes, scheduled tasks, services, and the SMB-client stack for tools such as PsExec, Impacket (smbexec.py / wmiexec.py), Cobalt Strike, or known worm payloads.

4. **Hunt for secondary infections**: although no peer-to-peer SMB *between other internal hosts* appears in this capture, the scan would have identified vulnerable

targets. Audit any host that responded with non-RST traffic (the small inbound flows from peers) for signs of compromise — credential dumps, new local admins, new services, scheduled tasks.

5. **Patch / verify MS17-010** and disable SMBv1 across the estate; segment workstation-to-workstation SMB at the firewall (MITRE T1021.002 mitigation guidance).

6. **Examine egress logs around the capture window** for the 1.16 MB outbound transfer to 200.10.122.10 — treat as a likely exfiltration incident and engage IR / notification procedures as required.

Bottom line: the graph does not show benign network activity. It shows one compromised internal host — 192.168.3.104 — performing a network-wide SMB sweep (T1046 + T1021.002) while simultaneously communicating with external infrastructure in a pattern consistent with C2 and data exfiltration. Treat this as an active intrusion until proven otherwise.

MC-210-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate the Neo4j database to analyze the network traffic for signs of malicious activity. Let me start by loading the necessary tools.

Searched available tools

Searched available tools

Let me first inspect the database schema to understand the structure of the network traffic model.

Get Neo4j Schema

Get Neo4j Schema

The schema tool has an issue with the default sample size. Let me provide an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The schema shows network entities: IPs, Ports, Hosts, OS, Domains, and geographic data. Notably, IP nodes have threat-intelligence flags (is_spam, is_known_attacker, is_Bot, is_Local). Let me explore the actual data.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the external IPs in detail — their geographic locations and the ports they communicate on.

Used neo4j-database integration

Used neo4j-database integration

Interesting — three external IPs (all in Cologne, Germany) communicate on port 5938. Let me see all ports and how local hosts connect to these external IPs.

Used neo4j-database integration

Used neo4j-database integration

The Host nodes are actually the external endpoints (all OS = "Cloud"). Let me examine how local IPs connect to ports, and the full relationship structure around the port-49xxx range.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a clear picture. The key finding is **port 5938** — used by local host 192.168.1.119 and by three external Cologne, Germany IPs. Let me verify the host details for the two active local IPs and check the IP-to-Host mapping for the local network.

Used neo4j-database integration

Used neo4j-database integration

I have a complete picture of the graph. Now let me query the MITRE ATT&CK matrix to map the observed behavior against known adversary techniques. The critical signal is the use of **TCP port 5938**.

Searched the web

Searched the web

Port 5938 is the signature port for **TeamViewer** remote-access software. Let me cross-reference this against the MITRE ATT&CK matrix — specifically the technique covering remote-access tools used by adversaries.

Searched the web

Searched the web

Let me check one more thing — whether the external IPs and psmanaged.com domain are actually associated with TeamViewer's official infrastructure, which is the deciding factor between benign use and abuse.

Searched the web

Searched the web

I have everything I need to produce a verdict. Let me present the analysis.

Verdict: No indications of malware or cyber-attack

Confidence: Moderate-to-high. The captured traffic shows benign, expected activity consistent with legitimate remote-support software. No threat indicators were found.

What the graph contains

The database models a small local network (192.168.1.0/24) with 21 IP nodes, 5 external "Host" endpoints, 31 ports, and geographic data. Of the local IPs, only two showed actual outbound port activity: 192.168.1.2 and 192.168.1.119. The rest were broadcast/multicast addresses (192.168.1.255, 224.0.0.x, 239.255.255.250, 0.0.0.0) — normal network noise.

Why this looks benign

1. Threat-intelligence flags are all clean. Every IP node carries is_spam, is_known_attacker, and is_Bot properties — and all 21 are set to false. None of the five external IPs is flagged as a known attacker, botnet member, or spam source.

2. The traffic signature is TeamViewer, used normally. The notable port in the capture is **TCP 5938**, which is used for TeamViewer, which is a software application used for remote control, desktop sharing, online meetings, and file transfer between computers. By default,

TeamViewer clients initiate outbound connections to the TeamViewer cloud infrastructure on port 5938/TCP & UDP. The pattern in the graph matches this exactly: local host 192.168.1.119 connects out on 5938 to three external IPs, and one of them — 92.51.156.90 — resolves to n92-51-156-90.cnet.psmanaged.com, hosted in **Cologne, Germany**. All three 5938 endpoints are in Cologne, which is consistent with TeamViewer's German-operated relay infrastructure (the company is headquartered in Göppingen, Germany). The remaining external traffic is plain **HTTPS on port 443** to IPs in Amsterdam and New York — ordinary web traffic. TCP UDP PortsWhatPortIs

3. The high-numbered ports are normal ephemeral ports. The 28 ports in the 49159–49186 range are Windows dynamic/ephemeral source ports — the OS assigns these automatically to outbound connections. Their presence is expected and not suspicious.

The MITRE ATT&CK consideration

In fairness, remote-access software does appear in the ATT&CK matrix. Technique **T1219 (Remote Access Software)** describes how an adversary may use legitimate remote access tools to establish an interactive command and control channel within a network, and TeamViewer is named explicitly among the abused tools. Threat actors have abused legitimate TeamViewer to maintain unauthorized remote access (e.g., the TeamSpy campaign), so unexpected outbound traffic to TCP 5938 can be an indicator of misuse. MITRE + 2

However, **mapping to a technique is not evidence of compromise.** T1219 is fundamentally about *dual-use legitimate software*. TeamViewer connecting outbound to TeamViewer's own relay servers — which is precisely what this graph shows — is the tool's normal, designed behavior. Genuine T1219 abuse would typically be accompanied by corroborating indicators: connections to attacker-controlled (non-vendor) endpoints, flagged malicious IPs, unusual install patterns, off-hours unattended sessions, or lateral movement. None of those appear here. The IP-reputation fields are clean, the destinations are vendor infrastructure, and there is no second technique in the chain.

Caveats and limits of this analysis

The verdict is bounded by what the graph captures. This model has no packet payloads, no timestamps, no traffic volumes, no process/parent-process data, and no DNS query logs. So I can confirm there are no *indicators* of attack in the modeled data, but I cannot fully rule out a sophisticated intrusion that left no trace in these fields. Two things would be worth a human analyst's confirmation: that the TeamViewer installation on 192.168.1.119 is authorized/sanctioned by the network owner (an unsanctioned RMM tool is a policy concern even when not malware), and that the 443 traffic to the New York and Amsterdam IPs corresponds to known-good services.

Bottom line: Based on the graph and the ATT&CK matrix, the captured traffic is consistent with legitimate use of TeamViewer plus ordinary HTTPS browsing. There are no flagged malicious IPs, no botnet/spam indicators, and no behavioral pattern that distinguishes this from normal remote-support activity. **No evidence of malware or cyber-attack** — with the standard recommendation to verify that the TeamViewer usage is sanctioned.

MC-210-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this network traffic graph for signs of malicious activity. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema inspection failed. Let me explore the database structure directly with Cypher queries.

Used neo4j-database integration

Used neo4j-database integration

Interesting — the IP nodes carry threat-intelligence flags. Let me look at the relationships and the actual data.

Used neo4j-database integration

Used neo4j-database integration

There are 5 external IPs. One — 92.51.156.90 — stands out. Let me look at the hosts, OS, domain, and the full port picture.

Used neo4j-database integration

Used neo4j-database integration

A clear pattern is emerging. Let me get the exact connectivity between the local hosts and external IPs, plus the full port detail on the busy host.

Used neo4j-database integration

Used neo4j-database integration

Now I have the full picture of the graph. The single most important fact is **port 5938** — that's a well-known fixed port. Let me also check which external IPs use the HTTPS connections and confirm the domain detail before consulting MITRE ATT&CK.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me consult the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me confirm the key fact about port 5938 — its specific association — and verify the domain pattern.

Searched the web

Searched the web

I now have everything needed for a verdict. Let me run two final confirming queries before I summarize.

Used neo4j-database integration

Used neo4j-database integration

Verdict: No indication of malware or cyber-attack

After querying the graph and cross-referencing against the MITRE ATT&CK matrix, my assessment is that **the captured traffic shows no signs of a compromise.** What initially looks suspicious is fully explained by legitimate remote-support software.

What the graph contains

The capture covers a small LAN (192.168.1.0/24) with ~10 local hosts plus the usual multicast/broadcast addresses. Only two local machines generated outbound traffic of note:

- **192.168.1.119** — connected to three external cloud-hosted IPs (162.220.223.28 in New York; 178.77.120.104 and 92.51.156.90 in Cologne, Germany), **all on the same port: 5938**.
- **192.168.1.2** — high local port activity (49159–49186), consistent with normal ephemeral client ports.

The two remaining external IPs (46.163.100.213, 168.63.124.19) were contacted only on **443 (HTTPS)** — ordinary encrypted web traffic.

Why this is benign

Port 5938 is the dedicated, IANA-registered port for TeamViewer, a legitimate remote-desktop product. By default, TeamViewer clients initiate outbound connections to the TeamViewer cloud infrastructure on port 5938/TCP & UDP, and where 5938 is unavailable it falls back to port 443 and then port 80. The observed pattern matches this exactly: WhatPortIsN-able

- Outbound-only connections from one host to **multiple geographically distributed cloud relay servers** on a single fixed port — this is precisely how TeamViewer's relay network operates. TeamViewer uses a network of more than 250 servers in data centers worldwide, with geolocation used to pick the nearest router. N-able
- The PTR hostname n92-51-156-90.cnet.psmanaged.com resolves to a hosting provider — consistent with a relay server, not a victim or C2 node.

Crucially, **every threat-intelligence flag in the graph is clean**: across all 21 IP nodes, is_known_attacker, is_Bot, and is_spam are all false. No flagged-malicious endpoint was contacted.

The MITRE ATT&CK consideration

The one technique worth weighing is **T1219 – Remote Access Tools (Software)**, under the Command-and-Control tactic. MITRE explicitly notes that adversaries abuse *legitimate* tools — remote access tools create a session between two trusted hosts… this software is commonly used for troubleshooting, software installation, and system management — and threat actors have abused legitimate TeamViewer to maintain unauthorized remote access, so unexpected outbound traffic to TCP 5938 can be an indicator of misuse. MITREInternet-Security

So the graph *does* show a technique that ATT&CK catalogs. But T1219 is a **dual-use** technique — the same traffic is generated by an IT helpdesk session and by an attacker. To call it malicious, you'd need corroborating indicators of an intrusion (initial-access activity, lateral movement, beaconing to flagged infrastructure, data staging). **None of those are present:** no flagged IPs, no internal host scanning patterns, no anomalous service ports, and only a single host using the RAT — exactly what authorized remote support looks like, not a compromise.

Bottom line

The traffic is **consistent with sanctioned use of TeamViewer for remote support on host 192.168.1.119**, plus routine HTTPS browsing. I find **no evidence of malware or a cyber-attack** in this capture.

One caveat worth stating: this verdict rests on the graph's threat-intelligence flags being accurate and complete. If you can confirm independently that TeamViewer is *not* authorized on 192.168.1.119, that single fact would flip the assessment — unsanctioned RAT installation is itself the finding. I'd recommend verifying that with whoever owns that machine.

MC-210-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

□□

□□

Claude responded: I'll investigate this network traffic graph for signs of malicious activity.

I'll investigate this network traffic graph for signs of malicious activity. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default parameter. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models hosts, IPs, ports, conversations, and geographic locations. Let me explore the data, starting with the IPs and their threat indicators.

Read Neo4j Cypher

Read Neo4j Cypher

The threat-intel flags (is_spam, is_known_attacker, is_bot) are all false, so I'll need to look at behavioral evidence. Let me examine the conversations, hosts, and the external IPs' geography.

Read Neo4j Cypher

Read Neo4j Cypher

One host — 192.168.1.119 — stands out: it talks to every external IP in the dataset. Let me look at the hosts, the ports involved, and the external IP geography.

Used neo4j-database integration

Used neo4j-database integration

Port **5938** is a very telling signal — that's the dedicated port for **TeamViewer**, a remote-access tool. It appears in 4 conversations, all involving host 192.168.1.119. Let me get the geography of the external IPs and confirm details.

Used neo4j-database integration

Used neo4j-database integration

Now I have a clear behavioral picture. Let me query the MITRE ATT&CK matrix to map these observations to documented adversary techniques.

Searched the web

Searched the web

Let me verify the TeamViewer/port 5938 association and check the destination IP hostname pattern, which I noticed is a managed-hosting/server domain.

Searched the web

Searched the web

I have enough to deliver a verdict. Let me do one final check — the conversation count per IP pair, to understand the connection pattern with the high-volume external host.

Read Neo4j Cypher

Read Neo4j Cypher

Verdict

No — there are no indications of malware or a cyber-attack in this capture. The traffic is consistent with **legitimate, sanctioned use of remote-support software (TeamViewer)** on a single workstation. Here's the reasoning.

What the graph shows

The capture covers a small Windows network (one domain, ~9 local hosts, the rest being broadcast/multicast addresses). Almost all traffic is mundane: SSDP discovery to 239.255.255.250, mDNS to 224.0.0.251, and DHCP/local broadcast. The only activity worth scrutinizing is **host 192.168.1.119**, which is the only local host that talks to the internet at all — to five external IPs in Germany, the Netherlands, and the US.

The decisive signal is **port 5938**. As confirmed against TeamViewer's own documentation, 5938 is the primary port TeamViewer uses, falling back to 443 and then 80 if it is unavailable, and it is essentially the only program known to use that port. The capture shows exactly this fingerprint on .119: port 5938 alongside HTTPS/443 and a large block of ephemeral high ports (49159–49186) — which matches TeamViewer's peer-to-peer fallback mechanism that dynamically selects available ports, with both devices initiating outbound connections. The heaviest external peer, 92.51.156.90, even resolves to n92-51-156-90.cnet.psmanaged.com — a managed-hosting/relay-server hostname, consistent with a TeamViewer routing server, not a residential or suspicious endpoint. TeamViewer + 2

MITRE ATT&CK mapping — and why it does *not* trigger here

This activity maps to **T1219 – Remote Access Tools** (and sub-technique **T1219.002 – Remote Desktop Software**). MITRE explicitly lists TeamViewer as software abused under this technique: an adversary may use legitimate remote access tools to establish an interactive command-and-control channel, software that is commonly used for troubleshooting, software installation, and system management. That dual-use nature is the whole point — T1219 is a *legitimate* tool that becomes malicious only in a malicious context. MITRE

The malicious context is **absent** here:

- **No threat-intel hits.** Every external IP has is_known_attacker, is_spam, and is_bot set to false. None is flagged.
- **Single host, not lateral spread.** A genuine compromise abusing RMM tools typically looks like deploying multiple RMM tools across many endpoints simultaneously for redundant persistent access. Here only one workstation uses TeamViewer; the other eight local hosts show no internet C2 traffic at all. Startupdefense
- **Traffic shape fits an interactive support session.** The .119↔92.51.156.90 flow is large and bidirectional (~1 MB up / ~677 KB down) — the signature of a live screen-sharing/remote-control session, not the small, periodic beaconing of malware C2 or the lopsided outbound spike of data exfiltration.
- **Internally consistent.** 192.168.1.2 (likely the gateway/server) mirrors the same ephemeral-port set with .119, exactly as expected when TeamViewer negotiates a P2P path through the LAN.

Caveats

This is a behavioral verdict based on a network-flow graph, so two honest limitations apply. First, the capture has no packet payloads or timestamps, so I can't distinguish a single long session from periodic beaconing with certainty — though the volume and bidirectionality strongly favor an interactive session. Second, T1219 is genuinely abused by real adversaries (Scattered Spider, Kimsuky, RTM and others). The benign verdict therefore rests on one assumption worth confirming with the network owner: **that TeamViewer is authorized software on host 192.168.1.119.** If it is not — if no one knowingly installed it — then this same traffic flips from "routine remote support" to "unauthorized remote-access channel" and would warrant investigation as a T1219 C2 incident.

Assuming TeamViewer is sanctioned (the overwhelmingly likely case given the clean threat intel, the single-host footprint, and the legitimate relay-server hostname): **the network shows no signs of malware or attack during the capture window.**

MC-210-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this for you. Let me start by exploring the Neo4j database to understand the network traffic model.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

Now I understand the structure. Let me explore the actual data — hosts, IPs, conversations, and ports.

Used neo4j-database integration

Used neo4j-database integration

Now I can see the structure: a local network (192.168.1.0/24) communicating with several external IPs. Let me examine the conversations in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at which ports are used in each conversation — this is critical for ATT&CK mapping.

Read Neo4j Cypher

Read Neo4j Cypher

A clear signal is emerging. Port **5938** appears repeatedly — that is the well-known TeamTracker/**TeamViewer** port. Let me confirm the per-IP port associations and check what ports each external IP uses.

Read Neo4j Cypher

Read Neo4j Cypher

The picture is now clear. The key finding is centered on host **192.168.1.119**. Let me now query the MITRE ATT&CK matrix to map these observations against documented adversary techniques.

Searched the web

Searched the web

Port 5938 is confirmed as TeamViewer. Let me verify the specifics of port 5938 and check whether the external IPs are TeamViewer infrastructure or something else.

Searched the web

Searched the web

Let me investigate one more thing — whether the n92-51-156-90.cnet.psmanaged.com host (the large data transfer endpoint) is TeamViewer infrastructure, since it does not match the *.teamviewer.com naming pattern.

Searched the web

Searched the web

I have enough to form a verdict. The psmanaged.com domain belongs to **PlusServer** (a German hosting provider — note ns1.plusserver.com in the SOA record), and cnet.psmanaged.com is a generic reverse-DNS pattern for IPs in their hosting range. The host n92-51-156-90 is simply the reverse-DNS encoding of IP 92.51.156.90. Let me do one final check on the loopback-source multicast traffic and the broadcast patterns to round out the picture.

Read Neo4j Cypher

Read Neo4j Cypher

I now have a complete picture. Here is my analysis and verdict.

Investigation Summary

The network model: The capture covers a small local network (192.168.1.0/24) with 8 internal IPs, a router/gateway (192.168.1.2), standard multicast/broadcast addresses, and 5 external hosts. All conversations are duplicated 2× in the graph (an artifact of how it was loaded — I treated each pair as one logical flow).

What the local hosts did: Most internal IPs (.110, .113, .117, .122, .123, .128, .129) only generate ordinary background noise — broadcast traffic to 192.168.1.255, LLMNR/mDNS to 224.0.0.251/252, and SSDP/UPnP discovery to 239.255.255.250. That is entirely benign.

The one host that talked to the internet: Only **192.168.1.119** communicated with external hosts. Its external conversations are:

External host	**Location**	**Port**	**Volume (sent / recv)**
168.63.124.19	Amsterdam, NL	443 (HTTPS)	29 KB / 305 KB
46.163.100.213	Cologne, DE	443 (HTTPS)	1 KB / 3 KB
178.77.120.104	Cologne, DE	**5938**	3 KB / 1 KB
162.220.223.28	New York, US	**5938**	0.4 KB / 0.6 KB
92.51.156.90 (psmanaged.com)	Cologne, DE	**5938**	**1.05 MB / 677 KB**

MITRE ATT&CK Mapping

Port **5938/TCP** is the dedicated, officially-assigned port for **TeamViewer** — no other programs are known for using the port, and TeamViewer primarily uses outbound TCP 5938 to connect to TeamViewer's routing/relay servers for remote access and screen sharing. Port LookupInternet-Security

This maps directly to **MITRE ATT&CK T1219 – Remote Access Software** (sub-technique **T1219.002 – Remote Desktop Software**), under the **Command and Control** tactic. ATT&CK explicitly lists this behavior: An adversary may use legitimate desktop support and remote access software, such as TeamViewer, Go2Assist, LogMein, AmmyyAdmin, etc., to establish an interactive command and control channel to target systems within networks, and notes that these tools give attackers the same capabilities as a user physically present at the computer, enabling them to execute commands, manipulate data, steal information, or deploy further payloads while bypassing typical network-based defenses. Center for Threat-Informed DefensePicus Security

Verdict: No reliable indication of malware or a cyber-attack — but one finding warrants human confirmation.

The decisive question is whether the TeamViewer traffic is *legitimate use* or *adversarial use*. The evidence points strongly toward **legitimate**:

1. **The endpoints are normal TeamViewer/hosting infrastructure, not malicious.** 92.51.156.90 resolves into psmanaged.com, which belongs to **PlusServer**, a mainstream German hosting provider (ns1.plusserver.com).
   The cnet.psmanaged.com hostname is just a generic reverse-DNS template encoding the IP itself (n92-51-156-90) — exactly the pattern of a TeamViewer relay/router server, not a bespoke attacker C2.
2. **Traffic shape matches an interactive remote-desktop session, not C2 beaconing.** A genuine attack via T1219 typically shows small, periodic "beacon" connections. Here the dominant flow is a single large bidirectional session (≈1.7 MB total) on the canonical port — consistent with someone actively running a TeamViewer screen-sharing session.
3. **No other attack indicators are present.** There is no scanning, no lateral movement between internal hosts, no suspicious internal-to-internal traffic, no exfiltration to anomalous destinations, no non-standard ports, and the other 443 traffic is ordinary HTTPS. The high-49xxx ports are just ephemeral client-side source ports, which is expected.

The honest caveat: TeamViewer is a *dual-use* tool. The same traffic is produced whether a help-desk technician (or the user) is running a sanctioned remote session, or an attacker has installed it for persistence/C2. The graph contains no information that distinguishes those two cases — no timestamps, no process data, no user identity, no indication of whether TeamViewer is sanctioned on host .119.

Recommended follow-up on host 192.168.1.119:

- Confirm with the user/owner whether TeamViewer is installed and authorized, and whether a remote session was expected during the capture window.

- Verify the installed TeamViewer binary against the vendor's published checksum — attackers (e.g., Kimsuky, RTM) deploy modified TeamViewer clients with pre-configured credentials, hidden UI elements, and auto-start configs, which have different file hashes than official releases. Startupdefense
- Check for unattended-access configuration and auto-start entries on that host.

Bottom line: The capture shows a single technique of interest — T1219 Remote Access Software on host 192.168.1.119 — but every contextual signal (legitimate hosting infrastructure, canonical port, interactive session profile, complete absence of scanning/lateral movement/exfiltration) indicates **sanctioned remote-support activity rather than an active compromise**. I'd classify this as **clean with one item to verify**, not as a detected attack.

MC-210-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this for you. Let me start by exploring the Neo4j database to understand the network traffic model.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

Now I understand the structure. Let me explore the actual data — hosts, IPs, conversations, and ports.

Used neo4j-database integration

Used neo4j-database integration

Now I can see the structure: a local network (192.168.1.0/24) communicating with several external IPs. Let me examine the conversations in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at which ports are used in each conversation — this is critical for ATT&CK mapping.

Read Neo4j Cypher

Read Neo4j Cypher

A clear signal is emerging. Port **5938** appears repeatedly — that is the well-known TeamTracker/**TeamViewer** port. Let me confirm the per-IP port associations and check what ports each external IP uses.

Read Neo4j Cypher

Read Neo4j Cypher

The picture is now clear. The key finding is centered on host **192.168.1.119**. Let me now query the MITRE ATT&CK matrix to map these observations against documented adversary techniques.

Searched the web

Searched the web

Port 5938 is confirmed as TeamViewer. Let me verify the specifics of port 5938 and check whether the external IPs are TeamViewer infrastructure or something else.

Searched the web

Searched the web

Let me investigate one more thing — whether the n92-51-156-90.cnet.psmanaged.com host (the large data transfer endpoint) is TeamViewer infrastructure, since it does not match the *.teamviewer.com naming pattern.

Searched the web

Searched the web

I have enough to form a verdict. The psmanaged.com domain belongs to **PlusServer** (a German hosting provider — note ns1.plusserver.com in the SOA record), and cnet.psmanaged.com is a generic reverse-DNS pattern for IPs in their hosting range. The host n92-51-156-90 is simply the reverse-DNS encoding of IP 92.51.156.90. Let me do one final check on the loopback-source multicast traffic and the broadcast patterns to round out the picture.

Read Neo4j Cypher

Read Neo4j Cypher

I now have a complete picture. Here is my analysis and verdict.

Investigation Summary

The network model: The capture covers a small local network (192.168.1.0/24) with 8 internal IPs, a router/gateway (192.168.1.2), standard multicast/broadcast addresses, and 5 external hosts. All conversations are duplicated 2× in the graph (an artifact of how it was loaded — I treated each pair as one logical flow).

What the local hosts did: Most internal IPs (.110, .113, .117, .122, .123, .128, .129) only generate ordinary background noise — broadcast traffic to 192.168.1.255, LLMNR/mDNS to 224.0.0.251/252, and SSDP/UPnP discovery to 239.255.255.250. That is entirely benign.

The one host that talked to the internet: Only **192.168.1.119** communicated with external hosts. Its external conversations are:

External host	**Location**	**Port**	**Volume (sent / recv)**
168.63.124.19	Amsterdam, NL	443 (HTTPS)	29 KB / 305 KB
46.163.100.213	Cologne, DE	443 (HTTPS)	1 KB / 3 KB
178.77.120.104	Cologne, DE	**5938**	3 KB / 1 KB
162.220.223.28	New York, US	**5938**	0.4 KB / 0.6 KB
92.51.156.90 (psmanaged.com)	Cologne, DE	**5938**	**1.05 MB / 677 KB**

MITRE ATT&CK Mapping

Port **5938/TCP** is the dedicated, officially-assigned port for **TeamViewer** — no other programs are known for using the port, and TeamViewer primarily uses outbound TCP 5938 to connect to TeamViewer's routing/relay servers for remote access and screen sharing. Port LookupInternet-Security

This maps directly to **MITRE ATT&CK T1219 – Remote Access Software** (sub-technique **T1219.002 – Remote Desktop Software**), under the **Command and Control** tactic. ATT&CK explicitly lists this behavior: An adversary may use legitimate desktop support and remote access software, such as TeamViewer, Go2Assist, LogMein, AmmyyAdmin, etc., to establish an interactive command and control channel to target systems within networks, and notes that these tools give attackers the same capabilities as a user physically present at the computer, enabling them to execute commands, manipulate data, steal information, or deploy further payloads while bypassing typical network-based defenses. Center for Threat-Informed DefensePicus Security

Verdict: No reliable indication of malware or a cyber-attack — but one finding warrants human confirmation.

The decisive question is whether the TeamViewer traffic is *legitimate use* or *adversarial use*. The evidence points strongly toward **legitimate**:

1. **The endpoints are normal TeamViewer/hosting infrastructure, not malicious.** 92.51.156.90 resolves into psmanaged.com, which belongs to **PlusServer**, a mainstream German hosting provider (ns1.plusserver.com).
   The cnet.psmanaged.com hostname is just a generic reverse-DNS template encoding the IP itself (n92-51-156-90) — exactly the pattern of a TeamViewer relay/router server, not a bespoke attacker C2.
2. **Traffic shape matches an interactive remote-desktop session, not C2 beaconing.** A genuine attack via T1219 typically shows small, periodic "beacon" connections. Here the dominant flow is a single large bidirectional session (≈1.7 MB total) on the canonical port — consistent with someone actively running a TeamViewer screen-sharing session.
3. **No other attack indicators are present.** There is no scanning, no lateral movement between internal hosts, no suspicious internal-to-internal traffic, no exfiltration to anomalous destinations, no non-standard ports, and the other 443 traffic is ordinary HTTPS. The high-49xxx ports are just ephemeral client-side source ports, which is expected.

The honest caveat: TeamViewer is a *dual-use* tool. The same traffic is produced whether a help-desk technician (or the user) is running a sanctioned remote session, or an attacker has installed it for persistence/C2. The graph contains no information that distinguishes those two cases — no timestamps, no process data, no user identity, no indication of whether TeamViewer is sanctioned on host .119.

Recommended follow-up on host 192.168.1.119:

- Confirm with the user/owner whether TeamViewer is installed and authorized, and whether a remote session was expected during the capture window.

- Verify the installed TeamViewer binary against the vendor's published checksum — attackers (e.g., Kimsuky, RTM) deploy modified TeamViewer clients with pre-configured credentials, hidden UI elements, and auto-start configs, which have different file hashes than official releases. Startupdefense
- Check for unattended-access configuration and auto-start entries on that host.

Bottom line: The capture shows a single technique of interest — T1219 Remote Access Software on host 192.168.1.119 — but every contextual signal (legitimate hosting infrastructure, canonical port, interactive session profile, complete absence of scanning/lateral movement/exfiltration) indicates **sanctioned remote-support activity rather than an active compromise**. I'd classify this as **clean with one item to verify**, not as a detected attack.

MC-210-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll analyze this network traffic graph for signs of malicious activity. Let me start by understanding the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models hosts, IPs, ports, conversations, and geographic locations. Let me explore the actual data, starting with the hosts and the network topology.

Used neo4j-database integration

Used neo4j-database integration

So the local network is 192.168.1.0/24 and there are five external (cloud-hosted) IPs in the US, Netherlands, and Germany. Let me examine the conversations in detail.

Read Neo4j Cypher

Read Neo4j Cypher

One host stands out clearly: 192.168.1.119 is the only local machine talking to all the external IPs. Let me look at the ports used in each conversation, which is the key to mapping behavior to ATT&CK techniques.

Read Neo4j Cypher

Read Neo4j Cypher

A very strong signal has emerged: **port 5938** appears in every external conversation involving 192.168.1.119. Port 5938 is the well-known signature port for **TeamRemote/TeamViewer-class remote access software**. Let me look at the full picture for that host and confirm the port detail.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me query the MITRE ATT&CK matrix to map this behavior to documented adversary techniques. Let me research the relevant technique.

Searched the web

Searched the web

This confirms the technique mapping. Let me check the bandwidth profile and the geographic spread one more time to fully characterize what 192.168.1.119 is doing, then deliver the verdict.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a verdict. Here is my analysis.

Verdict: Indications of a cyber-attack are present

There are clear indications of malicious activity, specifically a **remote access / command-and-control channel** on one host in the local network. The evidence is moderate-to-strong, not definitive (this graph contains only flow metadata — IPs, ports, byte counts — not packet contents or timestamps), but the pattern is characteristic enough to warrant treating 192.168.1.119 as compromised pending investigation.

What the graph shows

The local network is 192.168.1.0/24 with several workstations, a gateway/server at 192.168.1.2, and normal broadcast/multicast chatter (ARP, SSDP, mDNS, LLMNR to 224.0.0.x and 239.255.255.250). Most hosts only talk locally, which is benign.

One host is the outlier. 192.168.1.119 is the *only* internal machine communicating with the public internet, and it reaches **five external servers** hosted in three countries (Germany, Netherlands, USA). All of these are cloud-hosted ("Cloud" OS) endpoints rather than ordinary user-facing services.

Why this maps to a cyber-attack

The decisive indicator is **TCP port 5938**. It appears on four of the five external endpoints (92.51.156.90, 178.77.120.104, 162.220.223.28, and on the host itself). Port 5938 is the dedicated signature port of **TeamViewer-class remote access software**. This maps directly to:

- **MITRE ATT&CK T1219 – Remote Access Software** (Command & Control tactic). MITRE explicitly notes that key domains to monitor include *.teamviewer.com on TCP 5938, and that adversaries use legitimate remote access software such as TeamViewer, AnyDesk, and ScreenConnect to establish an interactive command and control channel to target systems. More specifically this is sub-technique **T1219.002 – Remote Desktop Software**. StartupdefenseMITRE

Several characteristics push this from "possibly legitimate IT support" toward "likely malicious":

- **Uncommon data flow direction.** MITRE's own detection guidance flags uncommon data flows, such as a client sending significantly more data than it receives from a server, as suspicious. On the largest channel (92.51.156.90), 192.168.1.119 *uploaded ~1.05 MB while downloading ~0.68 MB* — the client is sending more than it receives, consistent with **data exfiltration (ATT&CK Tactic TA0010)** rather than a normal

support session where the technician's screen-control traffic dominates inbound. Cyber-kill-chain

- **Multiple RMM relay endpoints from one host.** The host opened ~22 distinct ephemeral ports (49165–49186) to a single external server and contacted several relay IPs across regions. MITRE notes campaigns that generate connections to multiple RMM vendor endpoints from the same network, an abnormal pattern detectable through network flow correlation. Startupdefense
- **Possible secondary C2 channel.** Two other external IPs (168.63.124.19, 46.163.100.213) were reached over **port 443 (HTTPS)**. By itself HTTPS is normal, but on a host already showing RMM C2, an additional encrypted channel is consistent with **T1071 – Application Layer Protocol / Web Protocols** used as a redundant or fallback C2 path. The 168.63.124.19 flow pulled in ~304 KB inbound — consistent with a payload/tooling download.

Caveats

This conclusion rests on metadata only. Port 5938 *can* be legitimate — TeamViewer is widely used for genuine IT support, and MITRE stresses these tools are commonly used as legitimate technical support software and may be allowed by application control. The graph has no timestamps, so I cannot confirm whether sessions ran outside business hours (a key malicious indicator). The "more sent than received" pattern is suggestive of exfiltration but not conclusive. Cyber-kill-chain

Recommended response

Treat 192.168.1.119 as a suspected compromised host: isolate it from the network, verify whether TeamViewer/RMM software was authorized and installed by IT, inspect it for unrecognized remote-access tooling and persistence, and review the external IPs (92.51.156.90, 178.77.120.104, 46.163.100.213, 162.220.223.28, 168.63.124.19) against threat-intelligence feeds. The ~1 MB outbound transfer should be investigated specifically as potential data theft.

Bottom line: the capture shows a single internal host running remote-access software beaconing to multiple foreign cloud servers with an exfiltration-shaped traffic profile — a credible indication of an active C2 channel (MITRE ATT&CK T1219.002), not normal network behavior.

Cybersecurity is a sensitive area and metadata-only analysis has real limits — if this is a live incident, I'd recommend escalating to a security/IR professional who can examine the host and full packet captures directly.

NC-20-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

2:24 AM

Claude responded: I'll analyze the network traffic graph in Neo4j against the MITRE ATT&CK framework to detect potential cyber-attack indicators.

I'll analyze the network traffic graph in Neo4j against the MITRE ATT&CK framework to detect potential cyber-attack indicators. Let me start by exploring the database schema.

Used neo4j-database integration, loaded tools

Used neo4j-database integration, loaded tools

Good. I have a clear picture of the schema. Now let me understand the data better before mapping to MITRE ATT&CK.

The graph contains:

- **IPs** (725) with flags: is_Local, is_spam, is_known_attacker, is_Bot
- **Hosts** (711) belonging to IPs, with OS and Domain membership
- **Conversations** (729) between IPs with bytes sent/received
- **Ports** (3591) used in conversations
- **Geolocation** via City → Country

Let me start probing for indicators of compromise.

Read Neo4j Cypher

Read Neo4j Cypher

Significant findings already: 7 local hosts, 11 known attacker IPs, and 7 bot IPs. Let me see which local IPs are talking to these flagged IPs.

Used neo4j-database integration

Used neo4j-database integration

Now let me find direct conversations between local hosts and known attackers/bots:

Read Neo4j Cypher

Read Neo4j Cypher

This is extremely significant. The local host 10.0.2.15 has communicated with **all 18 flagged remote IPs** (11 known attackers + 7 bots). Let me characterize the host and the traffic patterns further.

Used neo4j-database integration

Used neo4j-database integration

So 10.0.2.15 is **the** workstation under observation. It is the source of basically all the network's outbound activity (728 conversations to 723 distinct remote endpoints). The other "local" addresses are broadcast/multicast/gateway/null artifacts. Let me look at the wider footprint, port distribution, and geographic spread.

Used neo4j-database integration

Used neo4j-database integration

Note: in this dataset bytes_sent and bytes_received are identical for every conversation, so this graph cannot tell us flow direction. I'll treat byte counts as total volume only. Let me check a couple more behaviors before mapping to MITRE ATT&CK.

Used neo4j-database integration

Used neo4j-database integration

DNS traffic of ~2.76 MB to the local resolver is wildly out of proportion to normal name resolution. This is a strong signature of **DNS tunneling / DNS-based C2**. Let me look at the domain set the workstation belongs to / contacts, and check for any pattern indicators.

Used neo4j-database integration

Used neo4j-database integration

This is interesting context. Several "known attacker" IPs are Google (1e100.net), Facebook (fbcdn.net), AOL (aol/152.163.x), or smtp.centrum.cz. Two 104.244.46.x IPs without rDNS are Twitter ranges. The "bots" map to Yandex and Yahoo crawlers / ad services. So the threat-intel labels here are noisy: many of these IPs are well-known major platforms that have appeared on some blocklists rather than being actual attack infrastructure. The smtp.centrum.cz contact, however, is notable since 10.0.2.15 transferred ~4.5 MB to it.

Let me check whether 10.0.2.15 also used SMTP-typical ports anywhere, and look at outbound port-25/465/587 use.

Read Neo4j Cypher

Read Neo4j Cypher

Only HTTP/HTTPS/DNS — no SMTP, RDP, SSH, SMB, FTP, IRC, etc. — so no obvious lateral movement or unusual service. The high-numbered "unknown" ports are simply ephemeral client-side ports for the HTTP/HTTPS/DNS sessions.

Used neo4j-database integration

Used neo4j-database integration

Wait — the byte values here don't match what I saw earlier (e.g. 104.244.46.7 showed 55.6 MB earlier). The discrepancy is because earlier I summed both legs of the conversation. Let me look more carefully — earlier I had bytes_sent: 55613602 for 104.244.46.7 but here only 5 MB. The reason is the earlier query was incorrectly counting via dual matches. Let me fix this:

Read Neo4j Cypher

Read Neo4j Cypher

So the convention is: sender → receiver, and bytes_sent = bytes_received because both ends of one conversation are stored together. So 5 MB inbound from 104.244.46.7. Let me redo the exfiltration-vs-download analysis properly:

Read Neo4j Cypher

Read Neo4j Cypher

Out: 6.5 MB. In: 80.5 MB. That ratio (≈1:12) is consistent with normal web browsing where the user mostly downloads pages, scripts, images, video. Not consistent with bulk exfiltration. Now let me find the conversations where the local host sent disproportionately MORE than it received — those are the suspicious ones.

Used neo4j-database integration

Used neo4j-database integration

OK, so each peer has either an outbound or an inbound conversation recorded — not both. This means the directional analysis is per peer. Let me find peers contacted with significant uploads where the IP is flagged:

Used neo4j-database integration

Used neo4j-database integration

Right — 104.244.46.7 / San Francisco / no real hostname → that IP block (104.244.42.0/23 etc.) belongs to Twitter/X. Categorizing it as "known attacker" is a fairly aggressive call from the threat-intel source. The 5 MB of inbound HTTPS over 11 ephemeral source ports is exactly what HTTP/2 keep-alive multiplexed connections to a Twitter feed look like.

Let me check one more set of indicators: long-tail / one-shot connections, and look at IRC/Tor/known C2 ports.

Used neo4j-database integration

Used neo4j-database integration

Negligible volume, no flags — likely just CDN edge nodes geolocated there. Now I have a complete picture. Let me research the MITRE ATT&CK techniques relevant to what I'm seeing.

Searched the web

Searched the web

Now I have enough technical grounding. Let me do one final cross-check: what happens if I look at the conversation patterns more holistically — is the traffic profile typical of a workstation that browses the web, or unusual?

Used neo4j-database integration

Used neo4j-database integration

Every large conversation is **inbound** — the local host is downloading from major web infrastructure (Fastly 151.101.x, Facebook, Akamai, Google 1e100.net, CloudFront, EdgeCast 192.229.x). All 16 large transfers are downloads, not uploads. The only "flagged" big peer (104.244.46.7 / Twitter) is also inbound.

I have everything I need to assemble the verdict.

Now I'll synthesize a verdict diagram and write up the analysis.

V

visualize

V

visualize show_widget

Verdict

Low-to-moderate suspicion of compromise — most likely benign, with one anomaly worth investigating.

The graph captures the activity of a single active workstation (10.0.2.15) that initiated 728 conversations to 723 external endpoints across 28 countries during the window. I mapped what the graph shows onto the MITRE ATT&CK enterprise matrix and ran the following checks.

What I found

Flagged IP contacts. The endpoint communicated with all 18 IPs labeled is_known_attacker or is_Bot in the graph. On its face this looks alarming, but resolving those IPs back to their hosts shows most are mainstream platform infrastructure: 1e100.net (Google), fbcdn.net (Facebook), 104.244.46.x (Twitter/X edge), yahoo.com, mds.yandex.net, yandex.ru. The only flagged peer outside that pattern is 46.255.231.70 (smtp.centrum.cz, a Czech mail provider), reached over HTTPS — not SMTP — with ~412 KB inbound, consistent with webmail use. This is the typical "noisy threat-intel feed" pattern: real reputational hits on shared infrastructure that produce many false positives.

DNS traffic to the local resolver (192.168.33.254) totals ~2.76 MB. This is the strongest single anomaly. Normal name resolution is on the order of hundreds of bytes per query; multi-megabyte DNS volumes are an indicator of MITRE **T1071.004 (Application Layer Protocol: DNS)** — DNS tunneling for C2 or low-and-slow exfiltration. However, the same signature can come from legitimate causes (heavily cached prefetching, anti-virus update lookups, or DNS-over-HTTPS through a local stub). The graph doesn't expose query labels or record types, so this can't be resolved on traffic counts alone.

No supporting indicators for an active attack. Across the rest of the graph the picture is normal web browsing:

- Only three well-known service ports are used: 80, 443, 53. No SMB, RDP, SSH, IRC, SMTP, or odd high-port services that would point to **T1021 (Remote Services)**, **T1571 (Non-Standard Port)**, or IRC-based C2.

- Total inbound bytes (80.5 MB) outweigh outbound (6.5 MB) by ~12×, the opposite of what **T1041 / T1567 (Exfiltration over C2 / Web Service)** typically produces. The largest outbound flow to any flagged IP is ~235 KB.

- All 16 conversations exceeding 1 MB are *inbound* HTTPS downloads from Fastly, Akamai, EdgeCast, CloudFront, Facebook, and Google — exactly what a workstation browsing modern websites looks like.

- The geographic spread (heavy US, plus EU CDN points-of-presence in Bulgaria, Hungary, Czech Republic, Austria) is what you get from any anycast CDN footprint, not contact with a specific adversary infrastructure.

Recommendation

Treat this capture as **clean by default, with one item to verify**: pull the raw DNS payloads exchanged with 192.168.33.254 and check for unusually long subdomain labels, high-entropy query strings, large TXT/NULL/MX responses, or sustained high query rates to a single second-level domain. If those are absent, the capture shows no evidence of malware or cyber-attack. If they're present, escalate as a likely T1071.004 case and quarantine 10.0.2.15 for endpoint forensics.

NC-20-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate the network traffic captured in the Neo4j database for indications of malicious activity. Let me start by loading the necessary tools and examining the database schema.

Searched available tools

Searched available tools

Let me examine the database schema first.

Get Neo4j Schema

Get Neo4j Schema

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

The graph models network traffic. Let me explore the data in more detail — starting with the structure of conversations and key indicators already flagged in the data.

Read Neo4j Cypher

Read Neo4j Cypher

Already interesting: 11 local IPs, 13 known-attacker IPs, and 8 bot IPs. Let me look at the local hosts and what they communicate with.

Read Neo4j Cypher

Read Neo4j Cypher

The local network is the 10.0.2.x range (a VirtualBox NAT-style subnet). The "local" set also includes broadcast/multicast addresses (224.0.0.x, 239.255.255.250, 255.255.255.255) and 0.0.0.0. The real endpoints of interest are 10.0.2.15, 10.0.2.2, 10.0.2.200. Let me look at the conversations involving the known attackers and bots.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me see which of these flagged IPs actually had conversations with the local hosts, and the data volumes involved.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the ports used in these conversations and the overall port profile, which is essential for ATT&CK mapping.

Read Neo4j Cypher

Read Neo4j Cypher

Important context emerging here. The "known attacker" / "bot" IPs actually belong to well-known internet services — 31.13.84.x is Facebook, 172.217.x.x is Google, 104.244.46.x is Twitter, 64.12.x / 152.163.x are AOL, 87.250.x / 213.180.x / 93.158.x is Yandex. The traffic is all on standard web ports (80/443) with normal ephemeral source ports. Let me look at the broader picture: all ports used by local hosts, and any non-standard listening ports.

Used neo4j-database integration

Used neo4j-database integration

Let me check the destination countries for all conversations, and look at the conversations on non-web ports (the null port and port 53/843/5100) and any spam-flagged IPs.

Used neo4j-database integration

Used neo4j-database integration

I have a comprehensive picture of the graph. Now let me query the MITRE ATT&CK matrix to ground the verdict in the actual framework. Let me set up a browser tab.

Searched the web

Searched the web

Let me run a few more targeted queries to check for the most diagnostic indicators of compromise before forming a verdict — beaconing patterns, the high-volume internal transfer, and any spam IPs.

Used neo4j-database integration

Used neo4j-database integration

I have everything I need. Let me present the verdict.

Verdict: No credible indication of malware or cyber-attack against the local network

After querying the graph and cross-referencing it against the MITRE ATT&CK framework, my assessment is that **the captured traffic represents ordinary client web browsing, not a compromise.** Here is the reasoning.

What the graph contains

The capture is from a small NAT'd subnet (10.0.2.0/24, a classic VirtualBox lab range). Only two real endpoints generated meaningful traffic — 10.0.2.15 (728 conversations) and 10.0.2.200 (286 conversations) — plus 10.0.2.2 (the gateway) and the usual broadcast/multicast addresses. There are 1,015 conversations to 986 remote IPs across 31 countries.

Why the "flagged" IPs are not a finding

The database tags 13 IPs as is_known_attacker and 8 as is_Bot. This looks alarming until you resolve them:

- 31.13.84.x → **Facebook**; 172.217.x / 216.58.x → **Google**; 104.244.46.x → **Twitter**; 64.12.x / 152.163.x → **AOL**; 87.250.x / 213.180.x / 93.158.x → **Yandex**; 217.12.15.54 / 77.238.185.x → **Yahoo**.

Every single one is the front-end infrastructure of a major, legitimate web service. The flags in the dataset are unreliable labels (likely from a stale or low-quality reputation feed), not evidence. Notably, is_spam is true for **zero** IPs, which is consistent with a benign capture.

What the ATT&CK framework would require — and what's absent

MITRE's **Command and Control (TA0011)** and **Exfiltration (TA0010)** tactics are where network-traffic compromise would show up. Diagnostic indicators for those — and their status here:

- **Beaconing / regular callbacks (T1071, T1571):** Not present. Conversations are one-off, of varied size, to ~986 *distinct* destinations — the fan-out pattern of a browser, the opposite of a C2 implant that talks repeatedly to a small number of hosts.
- **Non-standard ports (T1571):** Not present. All traffic is on **80/443** with normal high ephemeral source ports. No suspicious listeners, no IRC/4444/odd C2 ports.
- **Exfiltration asymmetry (T1041, T1048):** Not present. bytes_sent equals bytes_received in every record (the model stores a single volume figure), and the largest transfers are *inbound-heavy* downloads from CDNs (Akamai, Fastly 151.101.x, EdgeCast 192.229.x) — classic content delivery, not data theft.
- **Lateral movement (TA0008):** Not present. The one internal-to-internal flow (192.168.33.254 ↔ 10.0.2.15, ~2.7 MB) is between the host and its own VM-host gateway — expected in a virtualized setup, not SMB/RDP host-hopping.
- **Suspicious geography:** Russian (Yandex/Moscow) and Irish (Yahoo/Dublin) destinations exist, but they are name-brand services consistent with normal browsing, not bulletproof-hosting C2.
- **C2-over-web-service (T1102):** Worth noting only because attackers *can* abuse Twitter/Facebook/Google for C2 — but nothing here distinguishes this traffic from a person simply using those sites.

Caveats on confidence

This verdict is bounded by what the model captures. The graph has **no timestamps**, so I cannot test for periodic beaconing intervals — the strongest single C2 discriminator. It also has no payload, DNS query names, JA3/TLS fingerprints, or process attribution. A sophisticated adversary using domain fronting or legitimate-service C2 could in principle hide within this exact traffic shape. So the accurate statement is: **the data shows no indicators of compromise, and is positively consistent with benign browsing — but absence of evidence here is partly a limitation of the dataset.** If you want higher assurance, a capture with timestamps and DNS records would let me test for beaconing directly.

Bottom line: **benign.** The scary-looking attacker/bot flags are mislabels on mainstream web infrastructure, and none of the structural ATT&CK indicators for C2, exfiltration, or lateral movement are present.

NC-20-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate this for you. Let me start by examining the Neo4j database structure to understand what network traffic data is available.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection had an issue. Let me explore the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Now I have a clear picture of the graph. Let me examine the local network and look for suspicious indicators.

Read Neo4j Cypher

Read Neo4j Cypher

Significant findings already: 11 local IPs, 13 known attackers, and 8 bot IPs (all external). Let me examine the local network in detail.

Read Neo4j Cypher

Read Neo4j Cypher

The actual local hosts are 10.0.2.15, 10.0.2.200, and 192.168.33.254 (the rest are broadcast/multicast/gateway addresses). Let me check the attacker and bot IPs and how they relate to the local hosts.

Used neo4j-database integration

Used neo4j-database integration

Host 10.0.2.15 is heavily involved — communicating with 13 attacker IPs and 5 bot IPs. Let me look at the domains and hostnames these flagged IPs resolve to, and check the conversation/port patterns.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the "attacker"/"bot" flagged IPs resolve to legitimate services (Facebook, Google/1e100.net, Yahoo, Yandex, centrum.cz SMTP). This is a key nuance. The flags appear to come from a reputation/threat-intel feed, not necessarily an attack. Let me examine the actual traffic behavior — ports, conversation volumes, and what the local hosts are doing. Let me look at the port usage patterns.

Read Neo4j Cypher

Read Neo4j Cypher

The port 12350 stands out — that's an unusual high port appearing across 7 conversations, not an ephemeral source port range. Let me investigate it and other unusual ports.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 traffic is 10.0.2.200 talking to a cluster of 91.190.216.x / 91.190.218.x IPs. Let me check what those resolve to and look at the broader behavior of host 10.0.2.200.

Read Neo4j Cypher

Read Neo4j Cypher

These IPs have no reverse DNS (hostname = IP). The 91.190.216.0/24 block is associated with the Sality botnet / peer-to-peer malware infrastructure historically. Port 12350 + a cluster of no-DNS peers is a classic P2P botnet signature. Let me look at the domains queried and overall conversation profile of each local host.

Used neo4j-database integration

Used neo4j-database integration

Host 10.0.2.15 looks like normal heavy web browsing (large downloads from CDN/Fastly/Akamai/Facebook ranges). Host 10.0.2.200, by contrast, has the suspicious port-12350 P2P cluster. Let me characterize 10.0.2.200's contacts more closely and check for scanning/beaconing patterns.

Used neo4j-database integration

Used neo4j-database integration

A large number of tiny conversations (81 peers with <200 bytes) is a hallmark of scanning or P2P peer-discovery beaconing. Let me look at the tiny conversations and the port-12350 peers in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me query the MITRE ATT&CK matrix to map these observed behaviors to known adversary techniques. Let me search for the relevant technique details.

Searched the web

Searched the web

Let me do one more check on the data — examining whether the local hosts contacted any peers that look like a DGA (domain generation algorithm) or other discovery patterns, and the odd hostnames I saw earlier.

Used neo4j-database integration

Used neo4j-database integration

This is a notable finding. Those gooo...gle.cesnet.cz hostnames under the 195.113.214.x range are CESNET's well-known public network speed-test / measurement servers (the repeated "o" hostnames are a real CESNET naming convention), not a DGA — and the largest 10.0.2.200 flow (21.8 MB) went there, consistent with a speed test or large download. Let me do a final check on whether 10.0.2.200's port-12350 peers also appear with any other ports, to confirm the P2P interpretation.

Read Neo4j Cypher

Read Neo4j Cypher

I now have enough to produce a verdict. Let me give you the full analysis.

Verdict: Indications of malware are present — specifically a P2P-botnet infection on host 10.0.2.200

What the network looks like

The capture contains three real local hosts behind a small 10.0.2.0/24 (plus 192.168.33.254) network: 10.0.2.15, 10.0.2.200, and 192.168.33.254. The other "local" addresses (0.0.0.0, 224.0.0.x, 239.255.255.250, 255.255.255.255, 10.0.2.2, 10.0.2.255) are broadcast, multicast, and gateway addresses, not endpoints.

Host-by-host assessment

10.0.2.15 — Clean (benign heavy web browsing). Its 728 conversations and ~87 MB of traffic go almost entirely to ports 443/80 toward large CDN and service ranges (Fastly 151.101.x, Akamai, Edgecast 192.229.x, Facebook, Google). It does touch IPs flagged is_known_attacker/is_Bot, but those flags are misleading here — they resolve to *.1e100.net (Google), *.facebook.com, *.fbcdn.net, *.yahoo.com, and yandex.ru ad servers. Those flags come from a generic reputation feed and reflect ad/tracker reputation, not an attack. No technique signature.

192.168.33.254 — Clean. A single large conversation; nothing anomalous.

10.0.2.200 — Compromised. This is the finding. The evidence:

- A cluster of peers in 91.190.216.0/24 and 91.190.218.0/24 — IPs with **no reverse DNS**, no CDN/hosting identity, scattered across the address block — all communicating on **destination port 12350**, a fixed non-standard port that is not an ephemeral source port. 10.0.2.200 both initiates to and receives from these peers.
- A **long tail of ~81 tiny conversations** (under 200 bytes each) to distinct external IPs — the signature of peer-discovery / membership beaconing rather than normal traffic.
- The peer set behaves as a mesh: same fixed service port, mutual connections, no legitimate-service fingerprint.

This pattern — a fixed unusual C2 port shared across a dispersed set of no-identity peers, plus high-volume small-packet peer probing — is characteristic of **peer-to-peer botnet malware** (the 91.190.216.0/24 range has a documented history with P2P malware such as Sality).

MITRE ATT&CK mapping (10.0.2.200)

Tactic	Technique	Evidence in graph
Command & Control	**T1571 – Non-Standard Port**	C2/peer traffic on fixed port 12350
Command & Control	**T1090.001 – Internal/P2P Proxy**	mesh of peers, P2P-style trust relationships used to relay C2
Command & Control	**T1095 – Non-Application-Layer Protocol** (likely)	many flows with null/non-standard port records
Discovery	**T1018 – Remote System Discovery**	~81 tiny probe conversations enumerating peers

The 13 "known attacker" and 8 "bot" flagged IPs do **not** themselves constitute the attack — they are mainstream services with poor feed reputation. The genuine indicator is the structural anomaly: the port-12350 peer mesh and the beaconing tail on 10.0.2.200.

Recommendation

Treat 10.0.2.200 as infected: isolate it from the network, block egress to 91.190.216.0/24 and 91.190.218.0/24 and to port 12350, image the host for forensics, and inspect for autostart persistence. 10.0.2.15 and 192.168.33.254 show no signs of compromise.

A caveat on confidence: this verdict is built from flow metadata (peers, ports, byte counts) without packet payloads or timestamps, so it is a strong behavioral indication rather than a confirmed identification of a specific malware family. Payload/PCAP inspection of the port-12350 flows would confirm it definitively.

NC-20-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate this for you. Let me start by examining the Neo4j database to understand the network traffic model.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection failed. Let me query the schema directly.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type, starting with the key ones.

Used neo4j-database integration

Used neo4j-database integration

Let me understand the relationship structure and identify the local network hosts.

Used neo4j-database integration

Used neo4j-database integration

The 172.217.x.x addresses are actually public Google ranges (only 172.16.0.0–172.31.255.255 is private — 172.217.x is not). The genuinely local/private hosts are the 10.0.2.x range and 192.168.33.254. Let me focus on those.

Used neo4j-database integration

Used neo4j-database integration

Two active local hosts: 10.0.2.15 and 10.0.2.200. Let me look at where they're talking to and through which ports.

Used neo4j-database integration

Used neo4j-database integration

Now let me examine the ports being used — unusual ports are a key indicator. Let me look at non-standard ports used in conversations involving the local hosts.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 stands out — it's a non-ephemeral, non-standard port. The 49xxx range is normal ephemeral client ports. Let me investigate port 12350 in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 is used by host 10.0.2.200 communicating with a cluster of IPs in the 91.190.216.x/91.190.218.x range. This is a notable pattern. Let me investigate that IP block and the hosts/domains associated with it.

Read Neo4j Cypher

Read Neo4j Cypher

These IPs have no resolved hostnames (PTR records resolve only to the IP itself) and no domain — typical of peer-to-peer infrastructure. The 91.190.216.0 block has historically been associated with **Skype supernodes**. Let me check the broader context — what hostnames/domains the two local hosts connect to, and look for other suspicious indicators like DGA-style domains.

Read Neo4j Cypher

Read Neo4j Cypher

Two things jump out strongly. There's a cluster of bizarre hostnames in the cesnet.cz domain — gooooooooooogle.cesnet.cz, gooooooooooooogle.cesnet.cz, and one with ~50 o's: gooooo...oogle.cesnet.cz. These look like deliberate Google-impersonation / DGA-style names. And 10.0.2.200 connected to **195.113.214.204** (one of those hosts) with the single largest data transfer of any conversation (~21.8 MB). Let me examine these in detail.

Used neo4j-database integration

Used neo4j-database integration

Significant finding: 10.0.2.200 downloaded **~21.85 MB over plain HTTP (port 80)** from gooooooooooogle.cesnet.cz (195.113.214.204) — a host masquerading as Google but in the Czech academic network (cesnet.cz). Large HTTP downloads from a deceptively-named host strongly suggest a malware payload delivery. Let me check the second active host and other suspicious destinations, then query the MITRE ATT&CK matrix.

Read Neo4j Cypher

Read Neo4j Cypher

10.0.2.15's traffic looks like normal web browsing — Facebook, Akamai, Fastly (151.101.x), CloudFront CDNs over many ephemeral ports. The anomalous activity is concentrated on 10.0.2.200. Let me check the conversation count to those fake-Google hosts and look at DNS query patterns, then query the MITRE ATT&CK matrix.

Used neo4j-database integration

Used neo4j-database integration

Now let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me search for the ATT&CK techniques most directly tied to the patterns I found — fake domain names / masquerading and DNS tunneling.

Searched the web

Searched the web

I have enough from the MITRE ATT&CK matrix. Let me do a couple of final targeted queries to confirm the picture — checking the conversation count to the fake-Google host and whether there's any internal lateral movement.

Used neo4j-database integration

Used neo4j-database integration

I have enough to render a verdict. Let me lay out the findings visually.

Here's a verdict diagram mapping the network observations to the MITRE ATT&CK matrix.

Verdict: Yes — there are clear indications of a cyber-attack, localised to one host

After modelling the captured traffic against the MITRE ATT&CK matrix, the verdict is that **host 10.0.2.200 shows strong indicators of malware compromise.** The second active host, 10.0.2.15, looks clean.

What the evidence shows

1. Suspicious payload delivery (ATT&CK T1105 — Ingress Tool Transfer). The single largest conversation in the entire capture is a **~21.85 MB transfer over plain HTTP (port 80)** from 195.113.214.204 to 10.0.2.200. Adversaries may transfer tools or other files from an external system into a compromised environment. A multi-megabyte download over unencrypted HTTP from an oddly-named host is a textbook payload-delivery pattern. MITRE

2. Deceptive / DGA-style host naming (T1036 Masquerading + T1568.002 Domain Generation Algorithms). The download source is named **goooooooooooogle.cesnet.cz** — and there are two siblings, gooooooooooooogle.cesnet.cz and one with roughly 50 repeated o's. These impersonate Google but actually sit in the Czech academic network (cesnet.cz). DGAs can take the form of apparently random or "gibberish" strings when they construct domain names by generating each letter. Repeated-character "google" lookalikes are a recognised pattern in malware sandbox / C2 datasets. MITRE

3. Peer-to-peer style C2 over a non-standard port (T1571 / T1071). 10.0.2.200 used **port 12350** — a fixed, non-ephemeral, non-standard port — across 7 sessions to a tight cluster of IPs in 91.190.216.x / 91.190.218.x (Luxembourg/Ireland), none of which have resolvable hostnames or domains. Adversaries may communicate using application layer protocols associated with web traffic to avoid detection by blending in with existing traffic. This is consistent with peer-to-peer C2 beaconing. MITRE

4. The other host is a benign control. 10.0.2.15's 728 conversations are ordinary web browsing — Facebook, Akamai, Fastly, CloudFront CDNs over normal ephemeral client ports. No ATT&CK technique maps to it.

Caveats on confidence

Confidence is **moderate, not certain.** The graph models conversations, ports, and geolocation but lacks timestamps, packet/payload content, DNS query logs, and process attribution — so the verdict rests on behavioural correlation, not a confirmed malware sample. The bytes_sent and bytes_received fields are also always identical per conversation, which suggests the model stores a single volume figure rather than true directional byte counts. No internal lateral movement (T1570) was observed — the 10.0.2.x hosts barely talk to each other.

Recommended next steps

Isolate 10.0.2.200 from the network, capture full PCAP/payload of any further traffic to 195.113.214.* and 91.190.216/218.*, image the host for forensic analysis, and block those IP ranges and the *.cesnet.cz lookalike hostnames at the perimeter.

This is an analytical interpretation of a traffic model, not a definitive incident finding — a confirmed verdict would require examining the actual downloaded payload and the host itself.

NC-20-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll analyze the Neo4j database for indications of malicious activity. Let me start by loading the necessary tools and examining the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models network traffic
with IP, Host, OS, PORT, CONVERSATION, Domain, City, and Country nodes. Let me explore the data to understand what we're working with.

Read Neo4j Cypher

Read Neo4j Cypher

The conversations involve a local host 10.0.2.200. Let me identify the local network hosts and get a broader picture.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public Google IPs (not the private 172.16.0.0/12 range — only 172.16–172.31 is private). The genuine local/private network addresses are
the 10.0.2.x range and 192.168.33.254. The 10.0.2.x range is the classic VirtualBox NAT network. Let me confirm the actual local host.

Read Neo4j Cypher

Read Neo4j Cypher

Two local hosts dominate: 10.0.2.15 (728 conversations) and 10.0.2.200 (286). These are the machines under investigation. Let me get details on the hosts and their OS.

Used neo4j-database integration

Used neo4j-database integration

The host/OS data isn't tied to the local IPs in a useful way. The core evidence is in the conversations. Let me now analyze the traffic patterns of the two local hosts in depth — total volume, ports used, and destinations.

Read Neo4j Cypher

Read Neo4j Cypher

Note bytes_sent equals bytes_received in every conversation — the model stores total conversation volume rather than directional split. Let me look at the ports used across the conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Mostly HTTP/HTTPS, plus high ephemeral ports (49xxx, normal client-side) — and one notable anomaly: **port 12350**. That's a non-standard high port appearing in 7 conversations. Let me investigate port 12350 closely.

Read Neo4j Cypher

Read Neo4j Cypher

Significant finding. Host 10.0.2.200 is communicating on port **12350** with a cluster of IPs all in the 91.190.216.0/22 range. This is highly suspicious — port 12350 is a known C2 port. Let me geolocate these external IPs and check what else 10.0.2.200 talks to.

Read Neo4j Cypher

Read Neo4j Cypher

The 91.190.216.0/22 block belongs to **Skype/Microsoft's peer-to-peer infrastructure** (Luxembourg/Dublin Skype supernodes), and port 12350 is consistent with Skype P2P traffic. This particular cluster is likely benign. Let me set this aside and look more broadly. Let me examine the full set of external destinations and look for the classic ATT&CK indicators. First, let me query the MITRE ATT&CK matrix for relevant detection guidance.

Searched the web

Searched the web

I have the ATT&CK C2 detection guidance. Key network-observable indicators I should hunt for: **beaconing** (regular periodic connections to the same destination — T1071/T1571), **non-standard ports** (T1571), **connections to suspicious/foreign infrastructure**, **data exfiltration** (large outbound transfers — T1041/TA0010), and **lateral movement / internal scanning** (TA0008). Let me examine all external destinations of the two local hosts, grouped by geography.

Read Neo4j Cypher

Read Neo4j Cypher

Two things stand out. The **Czech Republic** received ~23.8 MB across only **30 conversations** — a very high bytes-per-conversation ratio compared to the US (513 conversations for 72 MB). And the US has 498 distinct IPs for 513 conversations — almost one conversation per IP, which is the normal "fan-out" of web browsing. The Czech traffic is concentrated. Let me drill into the Czech Republic destinations.

Read Neo4j Cypher

Read Neo4j Cypher

The single conversation 195.113.214.204 → 10.0.2.200 carries **~21.85 MB** — that's 57% of all of 10.0.2.200's traffic in one connection. The 195.113.x.x and 77.75.x.x ranges are Czech academic/Seznam.cz infrastructure (CESNET, the Czech academic network — this dataset is the well-known CTU-13 / Stratosphere capture from Czech Technical University). A large *inbound* transfer could be a legitimate download. Let me check the ports involved in that big transfer and look at the broader picture for 10.0.2.200.

Read Neo4j Cypher

Read Neo4j Cypher

The 21.8 MB transfer was over **port 80 (HTTP)** — a plain-HTTP download from a Czech academic server. Large but not inherently malicious (could be a software/ISO download). Now let me hunt for the strongest ATT&CK indicator: **beaconing** — repeated connections to the same external destination, which is the signature of C2. I'll look for external IPs contacted multiple times.

Read Neo4j Cypher

Read Neo4j Cypher

No external destination is contacted more than twice. There is **no repeated-beaconing pattern** — the model collapses repeat traffic into single CONVERSATION nodes (the count of "2" is mostly a sender-direction and receiver-direction pair for the same peer). The graph aggregates flows, so classic time-based beaconing can't be observed here. Let me check the internal traffic for lateral movement / scanning, then examine the noteworthy destinations (Russia, Iran, Iraq, Ukraine).

Read Neo4j Cypher

Read Neo4j Cypher

Internal traffic is minimal and benign: .255 is broadcast, 10.0.2.2 is the VirtualBox NAT gateway/DNS. **No host-to-host lateral movement, no internal port scanning** (no fan-out to many internal IPs). The two local hosts never even talk to each other. Now let me inspect the connections to higher-risk geographies.

Read Neo4j Cypher

Read Neo4j Cypher

This is very informative and actually reassuring. The "Russia" traffic is overwhelmingly to 77.88.x, 87.250.x, 213.180.x, 93.158.x — these are **Yandex** ranges (Russian search engine/mail). The "Bulgaria" traffic to 172.217.x is **Google** (GeoIP places some Google edge nodes in Sofia). The Iran IP 185.94.180.126 and the others are all over standard ports 80/443 with normal ephemeral source ports. None of this shows non-standard-port C2.

Let me do two final checks: look for any conversations on truly unusual ports (IRC, known-malware ports, etc.) and check for null-port / non-IP-protocol traffic.

Read Neo4j Cypher

Read Neo4j Cypher

Only three non-standard ports in the entire capture: **12350** and **33033** (both to the Skype/Microsoft Luxembourg range 91.190.216.x — consistent with Skype P2P client behavior), and **843** (Adobe Flash socket policy port — 52.91.6.144 is AWS). All have benign explanations. Let me do one last check on the null-port traffic and the conversation count totals for completeness.

Read Neo4j Cypher

Read Neo4j Cypher

The null-port traffic is all explainable: 8.8.8.8/8.8.4.4 (Google DNS), 224.0.0.251/252 and 239.255.255.250 (mDNS/SSDP multicast), 255.255.255.255 (DHCP broadcast), 64.4.23.x / 157.55.x (Microsoft/MSN). The one large item — 192.168.33.254 → 10.0.2.15 at 2.76 MB — comes from 192.168.33.254, which is likely the capture appliance / gateway interface.

I now have a complete picture. Let me give you the verdict.

Verdict: No indications of malware or cyber-attack

I queried the MITRE ATT&CK Command-and-Control guidance (TA0011) and systematically tested the captured traffic graph against each network-observable attack indicator. The capture itself is a small VirtualBox NAT lab network (10.0.2.0/24) with two active hosts — 10.0.2.15 (728 conversations, ~87 MB) and 10.0.2.200 (286 conversations, ~38 MB) — plus the gateway, broadcast, and a capture interface (192.168.33.254).

What I checked against ATT&CK, and what I found:

Non-standard port C2 (T1571). Traffic is 97%+ on ports 80/443. The only exceptions are port **12350/33033** (host 10.0.2.200 ↔ the 91.190.216.0/22 block) and port **843**. The 91.190.216.x block is Microsoft/Skype's peer-to-peer supernode infrastructure in Luxembourg and Dublin — 12350 is a normal Skype P2P port. Port 843 is the Adobe Flash socket-policy port. All have benign explanations.

Beaconing / periodic C2 (T1071, T1571). No external destination is contacted more than twice, and the apparent "2" counts are just paired send/receive flows to the same peer. There is no repeated, regular call-home pattern to any single endpoint. (Caveat below.)

Suspicious foreign infrastructure. Destinations in Russia, Iran, Iraq, and Ukraine appeared at first glance concerning, but resolve to ordinary services: the Russian IPs are almost entirely **Yandex** (search/mail), the "Bulgaria" 172.217.x IPs are **Google** edge nodes (GeoIP artifact), and the rest are normal web traffic on 80/443.

Data exfiltration (TA0010 / T1041). The largest single flow — ~21.8 MB *inbound* to 10.0.2.200 from a Czech academic network host (195.113.214.204, CESNET) over HTTP — is a download, not exfiltration. No anomalous large *outbound* transfer exists; outbound volumes are consistent with normal browsing.

Lateral movement / internal scanning (TA0008). Internal traffic is trivial — broadcast, DHCP, the NAT gateway. The two local hosts never even communicate with each other. No host-to-host probing or fan-out.

Important limitations of this conclusion:

This graph is an **aggregated flow model**, not a packet capture. It has no timestamps, so true time-based beaconing (e.g. a connection every 60 seconds) cannot be detected — repeat flows to one peer are collapsed into a single CONVERSATION node. It also stores bytes_sent equal to bytes_received for every flow, so directional asymmetry (a strong exfiltration signal) is unavailable. There is no DNS query content, no payload, no process/endpoint telemetry, and no threat-intelligence reputation data on the IPs. A low-and-slow C2 channel disguised as HTTPS to a reputable-looking host would not be visible at this resolution.

Bottom line: Within the limits of an aggregated flow graph, the traffic from 10.0.2.15 and 10.0.2.200 is consistent with ordinary user activity — web browsing, search, mail, Skype, and a large file download. I found **no evidence of C2, exfiltration, lateral movement, or scanning**. To raise confidence to a definitive "clean," you'd want the original PCAP with timestamps, DNS logs, and IP reputation enrichment.

NC-20-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate this. Let me start by exploring the Neo4j database structure to understand what network traffic data is available.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The APOC schema call failed. Let me explore the database structure manually.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type and the relationships.

Used neo4j-database integration

Used neo4j-database integration

Now I understand the graph structure. Let me identify the local network hosts and examine their traffic.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public Google IP ranges (not RFC1918 private space — 172.16.0.0/12 only covers 172.16–172.31, and 172.217 is outside that). The genuine local network is the 10.0.2.0/24 subnet. Let me examine those hosts.

Used neo4j-database integration

Used neo4j-database integration

The two active hosts in the local 10.0.2.0/24 network are 10.0.2.15 (728 conversations) and 10.0.2.200 (286 conversations); 10.0.2.2 is the gateway and 10.0.2.255 is the broadcast address. Let me profile the traffic of the two active hosts.

Used neo4j-database integration

Used neo4j-database integration

Note that bytes_sent and bytes_received are equal per conversation — likely a total-volume figure stored on both fields. Let me look at the destination ports used by the local hosts, which is a key indicator for MITRE ATT&CK mapping.

Used neo4j-database integration

Used neo4j-database integration

10.0.2.15 is associated with 3,587 distinct ports — that is an extremely high number and is a strong signal. Let me look at the well-known destination ports (the service ports being contacted, as opposed to ephemeral source ports).

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at all named (non-"unknown") ports across the whole capture to see if anything unusual stands out.

Used neo4j-database integration

Used neo4j-database integration

Let me look at the external peers that 10.0.2.15 contacts and their associated hostnames/domains — this is the key to spotting C2 (command-and-control) infrastructure or known-bad domains.

Read Neo4j Cypher

Read Neo4j Cypher

This is a large set of mostly mainstream destinations (Google, Facebook, Akamai, CloudFront, Amazon EC2, Yandex, Seznam — a Czech portal). I have the picture I need from the graph. Now let me consult the MITRE ATT&CK matrix to map observed behaviors against known adversary techniques.

Searched the web

Searched the web

I have a solid grounding in the relevant MITRE ATT&CK techniques. Let me run a few more targeted queries to check for the specific behavioral signatures these techniques would leave in the graph — particularly beaconing/repeated-contact patterns and any non-standard ports.

Read Neo4j Cypher

Read Neo4j Cypher

Each external peer is contacted in exactly one conversation — the graph aggregates traffic per IP-pair rather than recording timestamps, so there's no beaconing/periodicity dimension. Let me check the broadcast/multicast traffic and the destination ports used by the 10.0.2.200 host as well.

Read Neo4j Cypher

Read Neo4j Cypher

That's normal LAN noise — 224.0.0.252 is LLMNR, 255.255.255.255 and .255 are DHCP/NetBIOS broadcasts. Let me check whether the two local hosts ever talked to each other (potential lateral movement) and look at the country distribution of the external peers.

Used neo4j-database integration

Used neo4j-database integration

One peer is geolocated in Iran. Let me identify it and check its hostname, since traffic to certain jurisdictions can warrant a closer look.

Used neo4j-database integration

Used neo4j-database integration

That peer was contacted over standard HTTP/HTTPS with ordinary ephemeral source ports — a small (~15 KB) exchange, consistent with normal web browsing. Let me do one final check: confirm there are no destination service ports beyond 80/443/53 anywhere in the local hosts' traffic (the "unknown" ports are all high-numbered ephemeral client ports, which I'll verify).

Used neo4j-database integration

Used neo4j-database integration

The only privileged/service ports in the entire local-network traffic are 53 (DNS), 80 (HTTP), 443 (HTTPS), and 843. Port 843 is the legacy Adobe Flash policy-file port — benign for a 2015-era capture. Every other port is a high-numbered ephemeral client port. I now have a complete picture.

Verdict: No indications of malware or cyber-attack

I queried the Neo4j graph (986 IPs, 1,015 conversations, 3,596 ports) and cross-referenced the observed behavior against the relevant MITRE ATT&CK techniques. My assessment is **benign — no evidence of compromise**.

The local network. The genuine local subnet is 10.0.2.0/24 (a typical VirtualBox NAT range). It has two active hosts — 10.0.2.15 (728 conversations) and 10.0.2.200 (286) — plus the gateway 10.0.2.2 and broadcast .255. Note that the 172.217.x.x addresses are *not* private space; 172.16.0.0/12 ends at 172.31, so 172.217 is public Google IP space and not part of the local network.

What I checked against MITRE ATT&CK, and what I found:

Command & Control — Application Layer Protocol (T1071.001) and Non-Standard Port (T1571). The only service ports touched by local hosts are 53/DNS, 80/HTTP, 443/HTTPS, and 843 (legacy Adobe Flash policy port — benign). Everything else is high-numbered ephemeral client ports, exactly as expected for outbound browsing. No C2 over an unusual port, no service on a non-standard port.

C2 / Exfiltration Over C2 Channel (T1041) — beaconing. Every external peer appears in exactly one aggregated conversation. The graph stores per-IP-pair totals with no timestamps, so true periodicity can't be measured — but there is no fan-in to a single repeatedly-contacted endpoint, which is the structural footprint beaconing would leave even in aggregated data.

Destinations. The ~250 external peers resolve overwhelmingly to mainstream infrastructure — Google (1e100.net), Akamai, Amazon CloudFront/EC2, Facebook/fbcdn, Yandex, and Czech portals (Seznam, Zbozi) consistent with a Czech-region user. No algorithmically-generated domains, no known-malicious hosts, no suspicious dynamic-DNS or bulletproof-hosting patterns. One peer geolocates to Tehran, Iran (185.94.180.126), but it was a small (~15 KB) plain HTTP/HTTPS exchange — normal web traffic, not a red flag on its own.

Lateral Movement. Only two intra-LAN flows exist, both small and to/from the gateway (10.0.2.2) — normal. No host-to-host scanning or SMB/RDP/admin-port traffic between 10.0.2.15 and 10.0.2.200.

Data volumes. The ~174 MB and ~38 MB totals are spread across hundreds of normal web destinations, with no single large asymmetric upload that would suggest exfiltration. (Caveat: bytes_sent and bytes_received are identical per conversation, so directionality can't be analyzed — this limits exfiltration detection.)

Caveats on the data model. This graph is an aggregated traffic summary, not full packet capture. It has no timestamps (no beacon timing), no per-direction byte counts, and no payload — so it cannot detect encrypted-payload C2, slow low-volume beaconing, or DNS-tunneling content. Within those limits, nothing in the graph maps to a MITRE ATT&CK adversary technique. The traffic profile is that of one or two ordinary workstations browsing the web. **Verdict: no indications of malware or cyber-attack during the capture window.**

NC-24-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll analyze the network traffic graph in your Neo4j database against the MITRE ATT&CK framework. Let me start by loading the database tools and examining the schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models a network capture with IP nodes (some flagged as spam, bot, or known attacker), Host/OS/Domain nodes, PORT nodes, CONVERSATION nodes with byte counts, and geolocation. Let me explore the data systematically.

Read Neo4j Cypher

Read Neo4j Cypher

There are 7 local IPs, 10 known-attacker IPs, and 3 bot IPs. Let me identify the local network and the flagged external IPs.

Read Neo4j Cypher

Read Neo4j Cypher

The local network is the 10.0.2.0/24 subnet (a typical VirtualBox NAT range), with 10.0.2.15 being the host of interest, 10.0.2.2 the gateway, and broadcast/multicast addresses. Let me examine the known attackers and bots.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the conversations involving the local host 10.0.2.15 and any flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Let me focus specifically on conversations between the local host and the flagged IPs (attackers and bots), and examine ports used.

Read Neo4j Cypher

Read Neo4j Cypher

Every flagged IP communicated with the local host 10.0.2.15. Let me examine which ports are used in these conversations and across the host generally.

Used neo4j-database integration

Used neo4j-database integration

The local host 10.0.2.15 uses a long contiguous sequence of ephemeral ports (49157–54195), which is normal client behavior. Let me check what ports the local host *connects to* (the destination/service ports) and how many distinct external IPs it talks to.

Used neo4j-database integration

Used neo4j-database integration

Service ports are exclusively 80/443 — standard web traffic, no suspicious service ports (no IRC, no high-port C2 listeners, no RDP/SMB exfiltration ports). Let me check the geographic spread of destinations and look at the bot/attacker conversation details more closely, plus check whether any local host *received* inbound connections (a sign of a listener/backdoor).

Used neo4j-database integration

Used neo4j-database integration

Note the sender/receiver fields in CONVERSATION nodes appear to label the higher-volume side as "sender" — the byte counts are symmetric (sent == received), so directionality here isn't reliable for true initiator. What matters: the local host's outbound volume vs. inbound. Let me now query the MITRE ATT&CK matrix to map findings against known techniques.

Searched the web

Searched the web

Let me run a few more targeted queries to finalize the analysis — checking for beaconing patterns (many small, repetitive conversations to the same IP) and the conversation profile of the flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a verdict. No beaconing pattern exists — the maximum is 2 conversations to any single peer, and connections fan out across 361 distinct destinations. Let me compile the analysis.

Verdict: No reliable indication of malware or active cyber-attack on the local network during the capture

What the graph contains

The capture models a single active workstation, 10.0.2.15 (a VirtualBox NAT lab host — the 10.0.2.0/24 range and 10.0.2.2/10.0.2.15 addressing are the VirtualBox default), plus broadcast/multicast pseudo-addresses. It conducted 736 conversations with 361 distinct external IPs across 22 countries, totaling ~238 MB. Three external IPs are flagged is_Bot and ten are flagged is_known_attacker.

Mapping the observations to MITRE ATT&CK

The ATT&CK Enterprise matrix organizes adversary behavior into tactics running roughly Initial Access → Execution → Persistence → … → Command and Control → Exfiltration → Impact. Command and Control involves communicating with compromised systems, often by mimicking normal web traffic, and Exfiltration involves transferring stolen data out of the network. For a network-only capture like this one, the tactics that *would* leave a graph fingerprint are Command and Control (TA0011) and Exfiltration (TA0010). Here is what the graph shows against each plausible indicator: Trellix

Command and Control (TA0011) — not supported. The hallmark of C2 is repetitive, periodic "beaconing" to a fixed endpoint. Adversaries commonly attempt to mimic normal, expected traffic to communicate with systems under their control. The graph shows the opposite of a beacon: no external peer was contacted more than **twice**, and traffic fans out to 361 different destinations. There is no persistent callback host, no use of non-standard C2 ports (every service port is 80/HTTP or 443/HTTPS), and the local host's own ports are a normal contiguous block of ephemeral client ports (49157–54195). This is the signature of ordinary web browsing, not application-layer C2. MITRE

Exfiltration (TA0010) — not supported. Exfiltration techniques typically transfer data out of a target network over the command-and-control channel or an alternate channel. Exfiltration shows up as a heavily *outbound*-skewed transfer to an unusual destination. In this graph the largest transfers are all *inbound* downloads from Google/CDN ranges (the top three — 173.194.150.249, 216.58.201.110, 74.125.104.106 — are Google netblocks delivering 40 MB, 29 MB and 28 MB respectively). The byte counts are recorded symmetrically (bytes_sent == bytes_received), so the model doesn't even preserve true directionality, but the volume profile is consumption, not exfiltration. MITRE

Initial Access / Lateral Movement — no evidence. Only one local workstation is active; there is no internal host-to-host traffic to indicate lateral movement, and no inbound listener/backdoor port on the host.

About the flagged IPs

This is the one caveat. Every flagged IP did communicate with 10.0.2.15, but the contact is minor and benign-looking: all conversations are on ports 80/443, volumes are tiny (the largest, a "bot" at 87.250.250.119 in Moscow — actually Yandex's range — is ~106 KB; most attacker IPs exchanged only 1–35 KB), and each was contacted only once or twice. Crucially, the is_known_attacker/is_Bot labels appear to be coarse threat-intelligence/geo-IP enrichments rather than capture-derived evidence — several flagged IPs sit in Google (172.217.18.4), Facebook (31.13.84.4), Bing (40.77.226.250) and AOL (152.163.51.3) netblocks. These are almost certainly false positives from a stale or aggressive blocklist, not genuine adversary infrastructure. Brief, low-volume HTTPS contact with a handful of reputation-flagged-but-mainstream IPs is consistent with normal browsing (ad networks, CDNs, analytics).

Bottom line

The traffic profile is that of a single workstation doing normal web browsing and large media/CDN downloads. There is **no beaconing, no C2 channel, no exfiltration pattern, no lateral movement, and no anomalous ports or listeners** — i.e., no graph-level fingerprint of

any ATT&CK tactic. The threat-intel flags warrant a quick sanity check but, given their tiny volumes and the fact that several point at major legitimate services, they read as blocklist noise rather than compromise.

Two honest limitations: this dataset has no timestamps, so genuine *slow* beaconing (one callback per hour/day) cannot be fully excluded; and it has no packet payloads, DNS records, or process telemetry, so host-side compromise that produced no distinctive network shape would be invisible. If you want higher assurance, the next step would be to correlate against capture timestamps and the actual flagged-IP threat-intel sources. If you can tell me whether the CONVERSATION nodes have any time data I haven't seen, I can test the slow-beacon hypothesis directly.

NC-24-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework. Let me start by loading the necessary tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models a network capture with IP nodes (flagged for spam, known-attacker, bot status), Hosts, Operating Systems, Domains, Ports, Conversations, and geographic data. Let me explore the data.

Read Neo4j Cypher

Read Neo4j Cypher

Key finding already: 7 local IPs, 10 known-attacker IPs, 3 bot IPs, and no spam IPs. Let me identify the local hosts first.

Read Neo4j Cypher

Read Neo4j Cypher

The local network uses the 10.0.2.0/24 subnet (a typical VirtualBox NAT range). The real endpoint is 10.0.2.15; the others are infrastructure addresses (gateway 10.0.2.2, broadcasts, multicast). Let me look at the known-attacker and bot IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me examine the conversations between the local host and these flagged external IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Let me specifically look at conversations involving the flagged attacker and bot IPs, and the ports used.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me check the domains accessed and look at the overall port usage on the local host to understand the traffic profile.

Used neo4j-database integration

Used neo4j-database integration

Now let me check the destination IPs that the local host talked to on non-standard service ports (anything other than 80/443/53), which would be more suspicious for C2 traffic.

Read Neo4j Cypher

Read Neo4j Cypher

No traffic on unusual privileged service ports — only HTTP, HTTPS, and DNS. Let me check the geographic spread and whether there's anything resembling beaconing (many small, regular conversations to one destination). Let me look at the bot IP 87.250.250.119 more closely since it had substantial bidirectional traffic.

Used neo4j-database integration

Used neo4j-database integration

I have enough data from the graph. Now let me cross-reference these findings against the MITRE ATT&CK matrix to map the observed behaviors to known adversary techniques.

Searched the web

Searched the web

Let me look at the Command and Control tactic specifically, since that's the tactic most observable in network traffic.

Searched the web

Searched the web

I have what I need. Let me run two final queries to confirm details — the volume profile of the flagged-IP conversations and whether the local host received more than it sent overall (consistent with browsing vs. exfiltration).

Read Neo4j Cypher

Read Neo4j Cypher

I now have a complete picture. Here is my analysis and verdict.

Verdict: No clear indication of malware or a cyber-attack on the local network during the capture

The traffic is consistent with **normal end-user web browsing** by a single workstation. The "attacker" and "bot" flags in the graph are almost certainly **reputation-list false positives**, not evidence of compromise.

What the graph contains

The capture models one local host, 10.0.2.15 (a 10.0.2.0/24 VirtualBox-NAT subnet — the other six "local" IPs are gateway/broadcast/multicast addresses). It conducted **369**

conversations totalling ~203 MB, almost entirely over **ports 80, 443, and 53** — HTTP, HTTPS, and DNS. There is no traffic on unusual service ports.

How this maps to MITRE ATT&CK

The only ATT&CK tactic observable in pure netflow data like this is **Command and Control (TA0011)**, primarily technique **T1071.001 – Application Layer Protocol: Web Protocols**. Attackers use standard application protocols to disguise C2 traffic, with common mediums being HTTP, HTTPS, or SMTP. MITRE's own detection strategy looks for common application-layer protocols used by abnormal processes, with high outbound byte counts or irregular ports, possibly indicating command and control or data exfiltration. NETSCOUTMITRE

Critically, **none of those red flags are present here**:

- **Traffic direction is inbound-dominant.** The host *uploaded* only ~2.3 MB out of ~203 MB total. C2 and exfiltration are characterized by anomalous *outbound* volume; this is the opposite — it looks like content download (web pages, video, ads).
- **No irregular ports.** Everything rides on legitimate, expected web ports. The thousands of high-numbered ports in the graph are normal client-side **ephemeral source ports**, one per TCP connection — expected behavior, not C2.
- **No beaconing signature.** The flagged IPs were contacted in single, short, ordinary-looking conversations, not the regular low-volume heartbeat pattern typical of C2.
- **The flagged hosts are mainstream infrastructure.** The 10 "known attacker" IPs resolve to ranges and behaviors consistent with **Google (172.217.18.4), Facebook/Meta (31.13.84.4), Microsoft/Bing (40.77.226.250), Akamai/AOL CDN, and DNS-redirector services**. The 3 "bot" IPs are **Yandex (87.250.250.119, Moscow)** and two Irish hosting addresses — Yandex traffic over HTTPS is a normal search/portal visit. Together the flagged IPs account for just **0.25% of all traffic across 13 of 369 conversations** — far too peripheral to represent a compromise.

The destination domains reinforce
this: facebook.com, bbc.com, yahoo.com, seznam.cz, mail.ru, vk.com, europa.eu, plus CDN and ad-network domains (adform.net, dotomi.com, everesttech.net). This is the footprint of a person browsing news and social media (with a Czech/Russian-language lean), not a malware infection.

Caveats

This verdict is bounded by what netflow-level data can show. A graph of IPs, ports, byte counts, and conversations **cannot reveal** host-based tactics — Execution, Persistence, Privilege Escalation, Credential Access, or encrypted-payload C2 that perfectly mimics browsing. A truly stealthy implant using domain fronting or low-and-slow HTTPS beaconing would not necessarily stand out in this dataset. The is_known_attacker/is_Bot flags also appear to come from an unspecified threat-intelligence feed; such feeds routinely mislabel large cloud and CDN providers.

Bottom line: Based on the available evidence, there are **no indications of malware or a cyber-attack**. To raise confidence further, you would want endpoint logs, full packet payloads or TLS metadata (JA3/SNI), and DNS query content — none of which are in this graph.

NC-24-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework to assess whether there are indications of malicious activity. Let me start by understanding the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue. Let me explore the database structure manually.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type and the relationship structure.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — IP nodes have threat-intelligence flags. Let me look at the other node types.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand how the relationships connect these nodes.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a clear picture of the graph model. Let me identify the local network hosts first.

Read Neo4j Cypher

Read Neo4j Cypher

The local network is 10.0.2.0/24 with the primary host being 10.0.2.15 (a typical VirtualBox NAT default — likely the captured/monitored machine). Let me get an overview of the threat-intelligence flags across all IPs.

Read Neo4j Cypher

Read Neo4j Cypher

There are 10 known-attacker IPs and 3 bot IPs. Let me identify them and see which ones the local host communicated with.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me see the conversations between the local host and these flagged IPs, including data volumes and ports.

Read Neo4j Cypher

Read Neo4j Cypher

The local host 10.0.2.15 had conversations with all 13 flagged IPs. Let me enrich this — get hostnames, domains, geolocation, and ports for these flagged endpoints.

Read Neo4j Cypher

Read Neo4j Cypher

This is informative. Several "known attacker" IPs resolve to well-known CDN/ad-network infrastructure (adform.net, 1e100.net = Google, fbcdn.net = Facebook, dotomi.com, weebly.net). These threat-intel flags appear to come from reputation feeds that frequently flag ad/tracker infrastructure. Let me look at the ports and protocols used in these conversations, and check whether the local host is the one initiating outbound connections.

Read Neo4j Cypher

Read Neo4j Cypher

Traffic uses standard HTTP/HTTPS with high ephemeral source ports — normal web browsing patterns. Let me get the destination port distribution overall and check for any unusual destination ports (the markers of C2, scanning, or exfiltration).

Used neo4j-database integration

Used neo4j-database integration

Only ports 80, 443, and 53 are service ports; everything else is high ephemeral ports (49xxx–54xxx) — the client side of normal TCP sessions. No SSH, RDP, SMB, IRC, or other C2/lateral-movement ports. Let me check the data volume profile for signs of exfiltration, and look at the geographic spread.

Used neo4j-database integration

Used neo4j-database integration

Only 3 conversations are upload-heavy, and none are dramatically asymmetric (no gigabytes leaving with little coming back). The overall ratio is ~6 MB up vs ~197 MB down — a normal web-browsing/download profile. Let me check the geographic distribution and the destination countries.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the domains contacted — this often reveals the nature of the activity (legitimate web browsing vs. suspicious DGA-style or odd domains).

Read Neo4j Cypher

Read Neo4j Cypher

This confirms a clear picture: the traffic is dominated by CDNs (Akamai, CloudFront, Amazon AWS), Google infrastructure (1e100.net), ad/tracking networks (adform, adition, everesttech, turn.com, ibillboard), and ordinary content sites (bbc.co.uk, facebook, yahoo, mail.ru, europa.eu, Czech sites). This is the fingerprint of a web browser loading pages.

Let me do a couple more checks — verify there's no DNS-based exfiltration, no beaconing pattern, and check the host count anomaly (363 IPs but only 368 conversations).

Used neo4j-database integration

Used neo4j-database integration

That single non-local conversation is 0.0.0.0 → 255.255.255.255 — a normal DHCP broadcast. So essentially every conversation is the single host 10.0.2.15 talking to the internet. No internal host-to-host traffic, meaning no lateral movement. Let me run one final check on the conversations to known-attacker IPs against MITRE-relevant indicators, and confirm there are no unusual ports to those flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

I have everything I need. Let me now cross-reference these observations against the MITRE ATT&CK matrix to structure the verdict.

Searched the web

Searched the web

I now have everything needed to deliver a verdict. Here is my analysis.

Verdict: No Indication of Malware or Cyber-Attack

After querying the network-traffic graph and cross-referencing observed behaviors against the MITRE ATT&CK Enterprise matrix, the capture shows **normal client web-browsing activity with no credible indicators of compromise.**

What the graph contains

The capture models a single monitored host, 10.0.2.15 (a default VirtualBox NAT address), on the local 10.0.2.0/24 network. It participated in 368 of the 369 recorded conversations; the only other conversation is a benign 0.0.0.0 → 255.255.255.255 DHCP broadcast. There is **no internal host-to-host traffic at all.**

Findings mapped to MITRE ATT&CK tactics

Command & Control (TA0011) — not observed. Adversary C2 typically uses non-standard ports, raw IPs with no DNS, or regular "beaconing" to a fixed endpoint. Here, every service-side connection used only ports 80 (HTTP), 443 (HTTPS), and one DNS (53). There were no connections to SSH, RDP, SMB, Telnet, IRC, or other ports associated with C2 or remote-control tooling. Destinations resolve to mainstream infrastructure — Akamai, Amazon CloudFront/AWS, Google (1e100.net), Facebook (fbcdn.net), BBC, Yahoo, and ad/analytics networks — exactly the fan-out a browser produces loading web pages.

Exfiltration (TA0010) — not observed. Exfiltration shows as large, asymmetric *outbound* transfers. The host uploaded ~6 MB total and downloaded ~197 MB — a roughly 1:33 ratio, the normal signature of consuming web content. Only three conversations were upload-heavy, and even those were modest and roughly balanced (e.g., 161 KB up / 106 KB down). There is no DNS tunneling pattern (a single DNS conversation).

Lateral Movement (TA0008) — not observed. Zero internal connections means there is no propagation, no scanning of the local subnet, no SMB/RDP/WMI activity.

Reconnaissance / Initial Access / Execution / Persistence — no supporting evidence in network telemetry of this kind.

About the "known attacker" and "bot" flags

The IP nodes carry threat-intelligence flags: 10 marked is_known_attacker and 3 marked is_Bot. This is the only superficially alarming signal, but it does not hold up:

- The flagged IPs resolve to **legitimate, well-known infrastructure** — s1.adform.net, fra15s28-in-f4.1e100.net (Google), xx-fbcdn-shv-01-vie1.fbcdn.net (Facebook CDN), ams01-nessy-float2.dotomi.com, origin-cdn-sf2p.weebly.net, and mc.yandex.ru (Yandex Metrica analytics).
- All conversations with these IPs used **only ports 80 and 443**, and most were inbound-initiated content delivery.
- Reputation feeds routinely flag ad-tech, analytics, and shared-CDN IPs because that infrastructure is abused intermittently by others — a classic source of false positives. A genuine compromise would show these flagged IPs reached on odd ports with beaconing or asymmetric upload, which is not the case here.

Conclusion

The capture is consistent with a single user browsing the web. Traffic uses standard web ports, fans out to mainstream CDNs and content sites, and has a download-dominant volume profile. None of the MITRE ATT&CK network-observable tactics — C2, Exfiltration, Lateral Movement — have supporting evidence. The threat-intel flags are best explained as reputation false positives on ad/analytics/CDN infrastructure.

No indications of malware or cyber-attack on the local network during the capture window.

Two caveats worth noting: network metadata alone cannot rule out fileless or dormant malware that simply wasn't active during capture, and the dataset has no timestamps, so I could not test for periodic beaconing directly (though the port and volume profiles still argue against it). If you want higher assurance, host-based telemetry (process/persistence artifacts) would be the complementary check. I can also dig deeper into any specific conversation or IP if you'd like.

NC-24-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate this systematically. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

Good, I have the schema. The graph models network traffic: Hosts with IPs and OSes, in Domains, located in Cities/Countries, with Conversations over Ports. Let me explore the data to understand what's there.

Read Neo4j Cypher

Read Neo4j Cypher

The capture centers heavily on a single local host, 10.0.2.15 (a classic VirtualBox NAT address). Let me identify the local network hosts and the ports used.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public Google IPs (not the private 172.16–172.31 range). The genuine local/private hosts are 10.0.2.15 (the monitored host), 10.0.2.2 (NAT gateway), 10.0.2.255 (broadcast), and 192.168.33.254. Let me look at the ports used in conversations and the host details.

Read Neo4j Cypher

Read Neo4j Cypher

The high-numbered ports (53xxx) are ephemeral client ports. The real interest is in the destination ports and *which remote hosts* 10.0.2.15 talked to. Let me look at the destinations, their domains, and geographic distribution.

Used neo4j-database integration

Used neo4j-database integration

Now I have a good picture of the traffic. Before forming a verdict, I need to consult the MITRE ATT&CK matrix as instructed. Let me look at some specific suspicious patterns first, then query MITRE ATT&CK.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check what 10.0.2.15 *sends* the most (outbound/exfil indicators) and look at conversations on non-standard ports.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check conversations on unusual destination ports (not 80/443) and look at the IPs in higher-risk countries (Russia, China, Iran).

Used neo4j-database integration

Used neo4j-database integration

Now let me consult the MITRE ATT&CK matrix as the task instructs. I'll search the official MITRE ATT&CK knowledge base.

Searched the web

Searched the web

Let me fetch the relevant MITRE ATT&CK technique pages for the network-observable behaviors I need to evaluate against: Command and Control and Exfiltration tactics.

Searched the web

Searched the web

I have a solid grasp of the relevant MITRE ATT&CK tactics. Now let me run targeted queries against the graph to test for specific ATT&CK-aligned indicators. First, beaconing detection — repeated, regular conversations to the same destinations.

Used neo4j-database integration

Used neo4j-database integration

Let me check the IPs of greatest concern — the Iran IP and the Russian host.exepto.ru — and see what conversations they had with the local host.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the OS of the monitored host and look at the broadcast/scanning patterns, plus whether any internal host scanned others.

Used neo4j-database integration

Used neo4j-database integration

The OS data is generic ("Cloud" placeholder for all 353 hosts) — not informative. Let me check the conversation count distribution and look for any port-scan / sweep behavior (one host touching many ports or many destinations rapidly), plus the total unique remote IPs the local host contacted.

Used neo4j-database integration

Used neo4j-database integration

Let me look at the destination port distribution from the local host's perspective — whether traffic stays on standard web ports or hits unusual server ports (a key C2 indicator per ATT&CK T1571 Non-Standard Port).

Read Neo4j Cypher

Read Neo4j Cypher

This is a key finding: every server-side conversation uses only standard ports — 443, 80, and 53. No IRC, no non-standard high ports as *destinations*, no unusual protocols. Let me do two final checks: the data-volume ratio analysis (exfiltration indicator) and confirm there's no large asymmetric outbound transfer.

Read Neo4j Cypher

Read Neo4j Cypher

Only one mildly outbound-skewed conversation (≈97 KB sent vs 23 KB received to a Cloudflare IP on HTTPS) — entirely consistent with a normal web upload/form submission, not bulk exfiltration. The dominant flows are inbound-heavy (downloads). Let me do one final check on the DNS host to rule out DNS tunneling (ATT&CK T1071.004).

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to form a verdict. The DNS volume here (~1.2 MB total over the whole capture) is normal for a browsing session and shows no signs of tunneling. Let me present the complete analysis.

Verdict: No indications of malware or cyber-attack

After querying the network graph and evaluating it against the MITRE ATT&CK Enterprise matrix, **the traffic capture shows no indications of malware or a cyber-attack.** The data is consistent with a single workstation performing ordinary web browsing.

What the graph contains

The capture is built around one monitored local host, 10.0.2.15 (a standard VirtualBox NAT address), plus the NAT gateway 10.0.2.2, a broadcast address, and a DNS server 192.168.33.254. That host had 368 conversations with 361 unique remote peers, exchanging ~119 MB sent and ~85 MB received.

Evaluation against MITRE ATT&CK network-observable tactics

MITRE ATT&CK is a knowledge base of adversary tactics and techniques based on real-world observation. It maps observed adversary behaviors to specific tactics and techniques, and the network-observable ones are primarily **Command and Control (TA0011)** and **Exfiltration (TA0010)**. I tested the graph against each relevant technique: Palo Alto Networks

Command and Control — no evidence. C2 detection relies on spotting unusual traffic patterns, such as a sudden spike in data transfers to a rarely contacted IP, non-standard

ports, or non-application-layer protocols. The graph shows the opposite: every server-side conversation used only standard ports — 443 (HTTPS, 270 conversations), 80 (HTTP, 137), and 53 (DNS, 1). There is no IRC, no non-standard service port, and no non-application-layer protocol. This rules out T1571 (Non-Standard Port) and T1095 (Non-Application Layer Protocol). The destinations are overwhelmingly mainstream infrastructure — Google/1e100.net, Akamai, Amazon AWS, CloudFront, CDN77 — which is what you expect from a browser, not a beacon to attacker infrastructure. NETSCOUT

Beaconing — no evidence. Of 361 remote peers, only 8 appeared in more than one conversation, and each of those used different ephemeral client ports — the normal pattern of a browser reopening connections, not the fixed-interval, fixed-size callbacks characteristic of a C2 implant.

Exfiltration — no evidence. The flow ratios are strongly inbound-heavy (large downloads, small uploads), which is normal browsing. The single most outbound-skewed flow sent ~97 KB to a Cloudflare IP over HTTPS — trivial in size and consistent with a form post or upload, not bulk data theft. No conversation shows the large asymmetric outbound transfer that would suggest T1041 (Exfiltration Over C2 Channel).

DNS tunneling — no evidence. Total DNS traffic to 192.168.33.254 was ~1.2 MB across the whole capture, normal for a browsing session and far too low to indicate T1071.004 (DNS tunneling).

Reconnaissance/scanning — no evidence. No internal host contacted many peers on sequential ports or showed sweep behavior. The 1,713 ports on the local host are ephemeral source ports for outbound web connections — expected, not a scan.

Caveats on confidence

A few limitations are worth stating plainly. The graph models flow metadata only (endpoints, ports, byte counts) — it has no packet payloads, no timestamps, and no process/endpoint telemetry, so I can't analyze beaconing *intervals* or inspect encrypted content. ATT&CK itself notes that adversaries often mimic legitimate traffic over HTTPS or DNS specifically to evade network detection, so a sophisticated C2 channel hiding inside HTTPS to a reputable CDN cannot be 100% excluded from metadata alone. The presence of a few IPs in Russia, China, and one in Iran is **not** itself suspicious — they resolve to ordinary services (Yandex, Mail.ru, VK, Baidu) and the conversations with them were tiny and unremarkable. DFIR Global

Bottom line: Within the limits of flow-level data, nothing in this capture maps to a MITRE ATT&CK technique. The verdict is benign — ordinary web browsing by one host.

NC-24-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework. Let me start by loading the necessary tools.

Searched available tools

Searched available tools

Let me start by examining the database schema to understand the structure of the network traffic data.

Get Neo4j Schema

Get Neo4j Schema

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models network traffic with IPs, Hosts, OS, Ports, Conversations, Domains, and geographic data. Let me explore the data in detail, starting with the conversations and ports.

Read Neo4j Cypher

Read Neo4j Cypher

The local host appears to be 10.0.2.15 (a typical VirtualBox NAT address — likely a sandbox/analysis VM). Let me look at the local network hosts and the ports being used.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually Google public IPs (not RFC1918 private). The true local network is the 10.0.2.0/24 subnet (VirtualBox NAT). 10.0.2.15 is the VM/guest, 10.0.2.2 is the NAT gateway, 10.0.2.255 is broadcast. Let me examine the ports used by the local host and the conversations involving it.

Read Neo4j Cypher

Read Neo4j Cypher

That's a massive list of sequential ephemeral source ports (49157–54195) — over 2000 ports on a single host. That itself is a notable pattern. Let me now look at the destination ports actually used in conversations, and the domains contacted.

Read Neo4j Cypher

Read Neo4j Cypher

Traffic is overwhelmingly HTTPS/HTTP. Let me look at the domains contacted, and the geographic spread of the remote IPs.

Read Neo4j Cypher

Read Neo4j Cypher

The domains include a mix of legitimate CDNs and services (Akamai, Google/1e100, Amazon AWS, CloudFront, BBC, Facebook), Czech sites (seznam.cz, email.cz, casablanca.cz, im.cz — suggesting a Czech user/system), and ad networks. Let me look at the conversations the local host initiated (outbound), and check the high-volume outbound transfers more carefully — these matter most for exfiltration / C2 analysis.

Read Neo4j Cypher

Read Neo4j Cypher

The traffic ratios look like normal web browsing — outbound is small (KB-range requests), inbound is large (MB-range downloads). No large outbound exfiltration. Now let me check the geographic distribution and look specifically for any suspicious destinations — particularly the conversations not on ports 80/443.

Read Neo4j Cypher

Read Neo4j Cypher

I see — the "ports" listed against each conversation are the *ephemeral source ports* on the client side, not destination service ports. The actual service ports are 80/443/53 as seen earlier. This is normal browsing where each TCP connection uses a new ephemeral port. The one DNS server is 192.168.33.254 (port 53).

Now let me check the geographic distribution and look for IPs in countries that would be unusual, plus check the hostnames of remote IPs for anything suspicious.

Read Neo4j Cypher

Read Neo4j Cypher

A globally distributed set of destinations, consistent with web browsing (CDN edge nodes are spread worldwide). Let me look at the China and Iran/Russia destinations more closely, and check the hostnames associated with remote IPs to identify anything anomalous.

Read Neo4j Cypher

Read Neo4j Cypher

The China IPs (115.239.x, 119.146.74.x, 123.125.115.x, 180.97.93.x, 220.181.7.x) are Baidu/Tencent/QQ ranges — common web infrastructure. Russia includes Yandex, Mail.ru, VK — all mainstream. The Iran IP and host.exepto.ru are worth a closer look. Let me get details on those two, plus check the laiqiutiyuzhibo.com domain which stood out as an unusual one.

Read Neo4j Cypher

Read Neo4j Cypher

Small, balanced transfers — consistent with web requests, not C2 beaconing patterns or exfiltration. Let me check the ports/domains for these and the laiqiutiyuzhibo.com host, then look at the overall host inventory and OS.

Read Neo4j Cypher

Read Neo4j Cypher

rmkwdhcynmif.laiqiutiyuzhibo.com — a random-looking subdomain — is worth noting, but the parent domain translates roughly to a Chinese sports-streaming site, and the IP is OVH (France) hosting. Let me check the OS of the local host and the full host count, then look at the OS node and overall conversation count to confirm the scale.

Used neo4j-database integration

Used neo4j-database integration

Now let me check one more important angle — whether there's any direct communication between local hosts (lateral movement) or unusual port usage like SMB/RDP/IRC, and look at the conversations involving the broadcast and gateway addresses.

Read Neo4j Cypher

Read Neo4j Cypher

No SMB, RDP, SSH, Telnet, IRC, or known-malware ports. Let me do a final check on the conversations involving the gateway/broadcast and verify there's no second internal host (lateral movement) and check whether the DNS server traffic volume is normal.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough data from the graph. Now let me query the MITRE ATT&CK framework to map my observations against known adversary techniques.

Searched the web

Searched the web

Let me check the MITRE ATT&CK Command-and-Control tactic techniques most relevant to traffic-level analysis, to make sure my verdict maps the observable graph features correctly.

Searched the web

Searched the web

I now have everything needed for a verdict. Let me present the analysis with a visual summary first.

Here is a summary of my analysis, mapped against the MITRE ATT&CK framework.

Verdict: No indication of malware or cyber-attack

After querying the graph and mapping observations against the MITRE ATT&CK matrix, the capture shows **no signs of compromise**. The traffic is consistent with ordinary web browsing by a single user.

What the graph contains. One active local host, 10.0.2.15 (a VirtualBox NAT address — likely an analysis or workstation VM), talking to ~362 external IPs across 369 conversations. The other "local" addresses are just the NAT gateway (10.0.2.2), the broadcast address (10.0.2.255), and a DNS server (192.168.33.254 on port 53). The 172.217.x.x addresses that look private are actually Google's public range.

Why it maps to benign activity:

- **C2 / Application-layer protocol (T1071):** Service ports are almost exclusively 443 (HTTPS, 270 conversations), 80 (HTTP, 137), and 53 (DNS). The huge list of 49xxx–54xxx "ports" on the host are sequential *ephemeral client source ports* — exactly what a browser opening many short-lived connections produces, not a covert channel. There's no periodic beaconing pattern to a fixed endpoint.
- **Exfiltration (T1041 / T1048):** Outbound volume from the host is small (KB-range requests) and inbound is large (MB-range downloads) on essentially every conversation. That's the signature of *consuming* content, not *exfiltrating* it. No large asymmetric outbound transfer exists.
- **Non-standard / non-app-layer ports (T1571 / T1095):** No SMB, NetBIOS, RDP, SSH, Telnet, IRC, or common malware ports (e.g. 4444) appear anywhere.
- **Lateral movement (TA0008):** No host-to-host traffic within the 10.0.2.0/24 subnet — only gateway, broadcast, and DNS. A NAT VM with one host means there's no internal network to traverse anyway.
- **Destinations:** Domains resolve to mainstream infrastructure — Akamai, Google (1e100.net), Amazon AWS, CloudFront, BBC, Facebook — plus Czech-language sites (seznam.cz, email.cz, casablanca.cz), suggesting a Czech user. The destinations in China (Baidu/Tencent ranges) and Russia (Yandex, Mail.ru, VK) are normal web services, and the global geographic spread reflects CDN edge nodes, not adversary infrastructure.

One minor caveat. A few destinations have random-looking subdomains (e.g. rmkwdhcynmif.laiqiutiyuzhibo.com) reminiscent of Domain Generation Algorithms. However, those connections are tiny and balanced, the parent domains are real services, and CDNs/ad networks routinely use machine-generated hostnames — so this isn't a finding on its own.

Limitations of this analysis: the graph models flow metadata (endpoints, ports, byte counts) but not packet payloads, timestamps, or TLS certificate details. A determined adversary using low-and-slow C2 perfectly disguised as HTTPS would not necessarily show up in flow data alone. Based on everything the graph *does* contain, though, nothing suggests malicious activity during the capture window.

NC-24-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 23

I'll investigate this for you. Let me start by understanding the structure of the Neo4j database, then analyze the traffic for signs of malicious activity.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The APOC schema call needs a sample size. Let me provide one.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models a network capture with these entities:

- **IP** (363) → addresses, linked to Hosts, Cities, Ports, Conversations
- **Host** (353) → hostnames, with an OS and Domain membership
- **CONVERSATION** (369) → sender/receiver, bytes sent/received
- **PORT** (1716) → port numbers used in conversations
- **City/Country** (64/22) → geolocation of IPs
- **Domain** (68) → DNS/AD domains

Let me explore the actual data, starting with the network topology.

Read Neo4j Cypher

Read Neo4j Cypher

There's one obvious local host: 10.0.2.15 (a 10.0.2.x address, the default NAT range for VirtualBox — this is a captured VM). Let me look at the local network endpoints and the conversations they're involved in.

Read Neo4j Cypher

Read Neo4j Cypher

All conversations involve the single local host 10.0.2.15. Now let me look at the Hosts, Domains, and OS to understand what this machine is and what it's been talking to.

Read Neo4j Cypher

Read Neo4j Cypher

Most external hosts resolve to "Cloud" OS. Let me find the local host's own hostname/OS/domain and look at the domains it's been associated with.

Used neo4j-database integration

Used neo4j-database integration

This looks like ordinary web browsing — news (bbc.com, seznam.cz, europa.eu), social (facebook.com, vk.com, yahoo.com, mail.ru, yandex.ru), CDNs, and a lot of advertising/tracking domains. Now let me look at the **ports** the local host used, since unusual ports are a key indicator. Let me first examine the MITRE ATT&CK matrix to ground my analysis, then check ports.

Searched the web

Searched the web

I have the ATT&CK framework grounding. The network-observable tactics relevant to a passive traffic capture are primarily **Command and Control (TA0011)** and **Exfiltration (TA0010)**, plus possible **Initial Access** and **Discovery** indicators. Let me now examine the ports used by the local host, which is the strongest signal for C2 / non-standard channels.

Read Neo4j Cypher

Read Neo4j Cypher

These are all **ephemeral source ports** (49152–65535 range) — the OS-assigned client ports for outbound connections, one per connection. That's normal. The meaningful signal is the **destination/server ports** used in conversations. Let me query the ports attached to CONVERSATION nodes.

Read Neo4j Cypher

Read Neo4j Cypher

The destination-port picture is now clear. The only well-known service ports in use are **443 (HTTPS) — 270 conversations**, **80 (HTTP) — 137 conversations**, and **53 (DNS) — 1 conversation**. Everything else is ephemeral (49152–65535) — the *other* side of each TCP connection. There are **no** non-standard C2-style server ports (no 4444, 8080, 1337, IRC ports, Tor, etc.).

Let me do a few targeted checks against MITRE ATT&CK network-observable techniques: asymmetric data flows (potential exfiltration T1041/T1567), suspicious geolocations, and any internal-to-internal scanning (Discovery / Lateral Movement).

Read Neo4j Cypher

Read Neo4j Cypher

Only one conversation where the local host sent notably more than it received — 97 KB out vs 23 KB in to a Cloudflare IP (104.16.67.157). That's a trivially small volume, consistent with a normal HTTP POST / form upload, not bulk exfiltration. Let me check the geolocation spread and confirm there's no internal lateral movement.

Read Neo4j Cypher

Read Neo4j Cypher

A normal geographic spread for general web browsing — mostly US/EU CDN and cloud providers, with a Czech Republic concentration (matching seznam.cz, email.cz, im.cz domains — suggesting a Czech user). Let me verify there's no internal-to-internal traffic (scanning/lateral movement) and check the few "null"-port conversations.

Read Neo4j Cypher

Read Neo4j Cypher

The only "internal" traffic is entirely benign infrastructure: DHCP (0.0.0.0 → 255.255.255.255), the NAT gateway 10.0.2.2 (the VirtualBox default gateway/DNS), local subnet broadcast 10.0.2.255, and the LLMNR multicast address 224.0.0.252. No host-to-host scanning, no SMB/RDP/WinRM lateral movement. Let me do one last check — the very largest inbound transfers, to confirm they're legitimate content downloads.

Read Neo4j Cypher

Read Neo4j Cypher

That confirms it. The largest transfers are all inbound bulk downloads over **HTTPS/443** and **HTTP/80** from Google infrastructure (1e100.net — Google's domain, IPs 173.194.x, 216.58.x, 74.125.x) and shop.bbc.com. The biggest single flow is 56 MB downloaded from a Google server — entirely consistent with video streaming (YouTube) or a large file download. The byte ratios are heavily *inbound* (download-dominant), which is the **opposite** of an exfiltration profile.

I have everything I need for a verdict.

Verdict: No indications of malware or cyber-attack

The capture shows a single Windows host (10.0.2.15, a VirtualBox VM on the default 10.0.2.0/24 NAT network) doing ordinary web browsing. There are no indications of malware or a cyber-attack against the local network during the capture window.

What the graph contains

One local endpoint (10.0.2.15) plus its NAT gateway (10.0.2.2), talking to ~360 external IPs across 369 conversations. External peers geolocate to a normal US/EU CDN-and-cloud spread, with a Czech cluster (seznam.cz, email.cz, im.cz) suggesting a Czech-speaking user.

Evidence checked against MITRE ATT&CK (network-observable tactics)

Command & Control (TA0011) — not present. Every server-side port in use is a legitimate service: 443/HTTPS (270 conversations), 80/HTTP (137), 53/DNS (1). All other ports are ephemeral client ports (49152–65535), exactly as expected for outbound browser connections. There are no non-standard C2 ports, no IRC, no Tor, no beaconing port. The destinations are well-known web/CDN/ad infrastructure (Google 1e100.net, Cloudflare, Akamai, Fastly, Facebook, BBC, Seznam, Amazon), not unattributed hosts. This argues against Non-Standard Port (T1571) and Application Layer Protocol C2 (T1071).

Exfiltration (TA0010) — not present. Traffic is overwhelmingly download-dominant. The largest flows are bulk *inbound* transfers (56 MB and 40 MB from Google servers — consistent

with video streaming, 9 MB from shop.bbc.com). Only one conversation sent more than it received, and only 97 KB — a trivial volume consistent with a normal form/POST upload. No signature of T1041 / T1567 bulk exfiltration.

Discovery & Lateral Movement (TA0007 / TA0008) — not present. The only internal traffic is benign infrastructure: DHCP, the NAT gateway, subnet broadcast, and LLMNR multicast (224.0.0.252). No host-to-host scanning, no SMB/RDP/WinRM, no internal sweep.

Initial Access / Impact — nothing observable. No inbound connections to the host on listening service ports; the host is purely a client initiating outbound sessions.

Caveats

This is a verdict on what the dataset can show, and it has real limits. The model has no timestamps, no packet payloads, no TLS/JA3 fingerprints, no process attribution, and no DNS query log — so a *low-and-slow* C2 channel that perfectly mimics HTTPS to a legitimate-looking CDN, or malware delivered earlier and now dormant, would not necessarily appear here. The capture also covers only one host, so it says nothing about other machines on a real LAN. Within those limits, though, every indicator points to routine, benign web browsing.

If you want higher assurance, the things that would move the needle are DNS resolution logs (to spot algorithmically-generated C2 domains), connection timestamps (to detect regular beaconing intervals), and endpoint/process telemetry.

NC-26-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this. Let me start by understanding the structure of the Neo4j database.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema failed. Let me inspect the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me look at the properties of each node type.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand how the relationships connect things, and find the local network.

Used neo4j-database integration

Used neo4j-database integration

The local network has one real host: **10.0.2.15** (the others are gateway, broadcast, multicast, null addresses — typical of a single VM/sandbox capture on a 10.0.2.0/24 NAT network). Let me check for flagged external IPs and the OS.

Read Neo4j Cypher

Read Neo4j Cypher

Several flagged external IPs. Let me check the OS and the conversations involving the local host.

Used neo4j-database integration

Used neo4j-database integration

So 10.0.2.15 initiated 394 conversations. Let me look at the conversations with the flagged IPs and overall traffic patterns.

Read Neo4j Cypher

Read Neo4j Cypher

The local host communicated with all six flagged IPs. Let me get more detail — what ports, hosts/domains, and locations these flagged IPs use.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the "attacker"/"bot" labels here resolve to 1e100.net (Google), adform.net (ad network), and yahoo.com. These flags look like reputation-feed false positives or threat-intel noise rather than reliable indicators. The real signal needs to come from the *behavior*. Let me look at the port usage and the destinations the local host talked to.

Read Neo4j Cypher

Read Neo4j Cypher

Mostly 443/80 with ephemeral source ports — standard web traffic. Let me check the destination domains and look for anything suspicious (DGA-like, IRC/known C2 ports, large numbers of distinct destinations).

Read Neo4j Cypher

Read Neo4j Cypher

These destinations are all mainstream services (Google/1e100.net, Fortinet, booking.com, ad networks, Czech/Slovak hosting providers). Let me check geographic spread and look for low-byte "beacon-like" conversations and unusual ports.

Read Neo4j Cypher

Read Neo4j Cypher

All non-443/80 ports are ephemeral (49xxx–60xxx range). No suspicious service ports (no IRC 6667, no 4444, no DNS-over-nonstandard, etc.). Let me check the country spread and small "beacon" conversations.

Used neo4j-database integration

Used neo4j-database integration

Only a handful of small conversations, and no repeated identical-size beacons to one destination. Let me also check inbound conversations (anything connecting *to* the local host) and the count of distinct hosts the conversations resolve to.

Read Neo4j Cypher

Read Neo4j Cypher

I have a comprehensive picture of the traffic now. Before forming a verdict, let me query the MITRE ATT&CK matrix to map my findings against recognized adversary techniques.

Searched the web

Searched the web

Let me look at the MITRE ATT&CK techniques most relevant to network-traffic-level analysis — particularly Command and Control and Exfiltration tactics.

Searched the web

Searched the web

I now have enough to map findings against the ATT&CK framework. Let me run a couple of final confirmatory queries — checking for self-loop conversations (the odd 10.0.2.15→10.0.2.15 entries) and the overall distinct-destination count.

Read Neo4j Cypher

Read Neo4j Cypher

I have everything I need. Here is my analysis and verdict.

Verdict: No clear indication of malware or cyber-attack

Based on a full review of the captured traffic graph mapped against the MITRE ATT&CK framework, **the evidence does not support a finding of compromise.** The capture is consistent with a single workstation conducting normal web browsing.

What the graph contains

The "local network" is effectively a single host: **10.0.2.15** (a 10.0.2.0/24 NAT subnet — the textbook layout of a VM/sandbox). The other "local" addresses are infrastructure (gateway 10.0.2.2, broadcast 255.255.255.255/10.0.2.255, multicast 224.0.0.252, null 0.0.0.0). That host had 394 conversations with 199 distinct external destinations across 16 countries.

How I checked it against MITRE ATT&CK

I focused on the tactics that are actually observable at the netflow/conversation level — primarily **Command and Control (TA0011)** and **Exfiltration (TA0010)** — since the graph has no process, file, or credential data to assess Execution, Persistence, or Privilege Escalation.

Command and Control — Application Layer Protocol (T1071.001): The relevant indicators are connections to suspicious domains, anomalous data volumes, and beacon-like periodic callbacks. I found none. 416 of the conversations used ports 443/80; every other port was an ephemeral client port (49000–60000). There were no IRC, non-standard, or known-C2 service ports. The destinations resolve to mainstream infrastructure — Google (1e100.net), Fortinet, booking.com, Yahoo, ad networks, and Czech/Slovak hosting providers. I checked specifically for low-byte repeated "beacon" conversations to a single host and found only four small conversations, none repeating — not a beaconing pattern.

Exfiltration (TA0010): 192 of 394 conversations were upload-heavy, which initially looks notable, but the volumes and destinations explain it: the largest outbound flow (~9 MB) goes to a Facebook IP (31.13.92.x, Ireland), and other big uploads go to Google and CERN (188.184.x = CERN, Switzerland). This is the normal request/upload side of web sessions, photo/content uploads, and TLS overhead — not bulk transfer to an anomalous endpoint.

On the "flagged" IPs

Six external IPs carry is_known_attacker or is_Bot flags. **I'd treat these as threat-intel noise, not evidence**, because they de-anonymize to entirely benign services: the "attacker" IPs are 1e100.net (Google) and adform.net (a mainstream ad network), and all four "bot" IPs resolve to unknown.yahoo.com (Yahoo). The traffic to them is unremarkable HTTPS. These flags appear to be stale or low-confidence reputation-feed entries — exactly the kind of false positive that should be corroborated by behavior before being acted on, and the behavior here is clean.

Important caveats

This is a behavioral verdict from netflow-level metadata only. The graph has **no timestamps**, so I could not test for periodic beaconing in the time domain — only for volume/structure patterns. It also has **no payload, DNS query, process, or host telemetry**, so techniques like DNS tunneling, encrypted-payload C2, or anything in the Execution/Persistence/Discovery tactics simply aren't visible in this dataset. "No indication" means no indication *within what was captured* — it is not a guarantee of a clean host. For a definitive judgment you'd want endpoint logs and timestamped flow records.

Bottom line: nothing in this capture rises to a credible indicator of compromise; it reads as routine web activity from one machine.

NC-26-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this. Let me start by loading the Neo4j tools and exploring the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC isn't available with the default settings. Let me explore the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me inspect the properties of each node type, starting with the key ones.

Used neo4j-database integration

Used neo4j-database integration

Let me look at the relationship structure to understand how everything connects.

Used neo4j-database integration

Used neo4j-database integration

Now let me identify the local network hosts — the computers we're investigating.

Read Neo4j Cypher

Read Neo4j Cypher

There's a single real local host: 10.0.2.15 (the others are broadcast/multicast/gateway/unspecified addresses). This is a classic single-host capture. Let me check which external IPs are flagged.

Read Neo4j Cypher

Read Neo4j Cypher

Several flagged external IPs. Let me examine the conversations the local host had with each of these, and the overall traffic profile.

Read Neo4j Cypher

Read Neo4j Cypher

The local host 10.0.2.15 communicated with all six flagged IPs. Let me identify the hostnames/domains behind those IPs, and check the ports involved.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the "flagged" IPs resolve to Yahoo, Adform (an ad-tech company), and Google (1e100.net). Let me check the property values more carefully and look at the ports used in these conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Let me get a broader picture of the local host's overall traffic — total conversations, top destinations by volume, and ports used.

Used neo4j-database integration

Used neo4j-database integration

Now let me check the MITRE ATT&CK matrix for technique context. Let me search the web for the framework's relevant tactics around network indicators.

Searched the web

Searched the web

Let me look more closely at the network-relevant ATT&CK techniques and check the data for specific indicators. First, let me examine the conversations with the most heavily-flagged behaviors — outbound-heavy data transfers and beacon-like patterns.

Used neo4j-database integration

Used neo4j-database integration

Let me check the country distribution and any unusual ports across all the local host's traffic.

Read Neo4j Cypher

Read Neo4j Cypher

All server-side ports are 443 (HTTPS) and 80 (HTTP) — no IRC, no SSH, no RDP, no SMB, no known C2 ports. Let me check the destination countries broadly and look for the "Cloud" OS detail.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the domains contacted to assess whether any look like malicious infrastructure (DGA-style, suspicious TLDs, etc.).

Read Neo4j Cypher

Read Neo4j Cypher

This is very informative. The domain list is overwhelmingly legitimate, recognizable infrastructure: CDNs (Akamai, CloudFront), cloud providers (AWS, Google 1e100.net/googleusercontent), ad networks (Adform, Adition, AppNexus/adnexus, Turn), Czech consumer services (Seznam.cz, Post.cz, Zbozi.cz, CEZ), Booking.com, Walmart,

Yahoo, Microsoft Edge (a-msedge.net), and CERN (cern.ch). No DGA-style random domains, no suspicious TLDs, no obvious C2 infrastructure.

Let me check whether 10.0.2.15 ever acts as a server (receiving inbound connections — possible scanning/lateral movement), and look at the conversation count distribution.

Read Neo4j Cypher

Read Neo4j Cypher

This confirms the pattern: every "inbound" conversation involves the external party on port 443/80 and the local host on high ephemeral ports (49xxx–51xxx) — meaning 10.0.2.15 is the **client** in every case. It initiated all connections. It never acted as a server, so there's no sign of it being scanned, exploited remotely, or used for lateral movement/hosting C2.

Let me do a couple of final checks — the "31.13.92.x" range (the heaviest upload destination) and whether any conversation has a beacon-like tiny-byte signature.

Read Neo4j Cypher

Read Neo4j Cypher

The 31.13.92.0/24 range belongs to Facebook (a well-known Facebook/Meta CDN block in Ireland) — the large uploads there are consistent with uploading photos/video to Facebook. Let me do a final check for small, beacon-like conversations that could indicate C2 check-ins.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to produce a verdict. Let me compile the analysis.

Verdict: No indications of malware or cyber-attack

After querying the graph and cross-referencing against the MITRE ATT&CK framework, my assessment is that **the capture shows no credible evidence of malware infection or a cyber-attack** affecting the local network.

What the network looks like

The capture contains a single real local host, 10.0.2.15 (the other "local" addresses — 0.0.0.0, 10.0.2.2, 10.0.2.255, 224.0.0.252, 255.255.255.255 — are the unspecified address, gateway, broadcast and multicast addresses). The 10.0.2.x addressing and the single OS node labeled "Cloud" indicate this is a VM-based capture, likely a sandbox or analysis environment. The host had 394 conversations with ~389 external IPs across 19 countries, exchanging ~34 MB out / ~78 MB in.

Why the "flagged" IPs are not convincing

Six external IPs carry is_Bot or is_known_attacker flags. However, when resolved through the graph, every one of them belongs to mainstream, legitimate infrastructure:

- 216.58.205.238 and the 1e100.net hosts → **Google**
- 217.12.15.83, 77.238.185.35, 217.12.13.40/41 (unknown.yahoo.com) → **Yahoo**
- 37.157.6.253 (s1.adform.net) → **Adform**, a mainstream ad-tech company

These flags almost certainly reflect over-broad threat-intel labeling (ad networks and large mail/CDN providers are frequently tagged as "bot" infrastructure because automated traffic transits them). The traffic to them is ordinary HTTPS, not attacker behavior.

Mapping against MITRE ATT&CK

Checking the graph for the network-observable tactics — Command and Control (TA0011), Exfiltration (TA0010), Initial Access, Lateral Movement, Discovery — none of the characteristic signatures are present:

- **Server-side ports are exclusively 443 (HTTPS) and 80 (HTTP).** No IRC (6667), no SSH/RDP/SMB, no common C2 ports (4444, 8443, 1080, 9001). This argues against a *Non-Standard Port* (T1571) or *Application Layer Protocol* abuse pattern.
- **The local host is the client in every single conversation** — external parties always sit on 443/80, the local host always on ephemeral 49xxx–51xxx ports. It never accepts inbound connections, so there is no sign of remote exploitation, hosted C2, or *Lateral Movement* to other internal machines (there are no other internal machines).
- **Destinations are recognizable, benign services**: CDNs (Akamai, CloudFront), cloud (AWS, Google), Microsoft Edge services, Booking.com, Walmart, CERN, and Czech consumer sites (Seznam.cz, Post.cz, CEZ). There are **no DGA-style random domains, no suspicious TLDs**, which weaks the case for *Dynamic Resolution* (T1568) C2.
- **No beaconing signature**: large transfers go to Facebook's 31.13.92.0/24 block (consistent with uploading media) and to Google/CERN — not small, regular check-ins to a single odd endpoint.
- The largest "upload" (~9 MB to Facebook) is normal social-media usage, not *Exfiltration Over C2 Channel* (T1041) to attacker infrastructure.

Caveats

This verdict is bounded by what the graph contains. The model has **no timestamps, no packet payloads, no DNS query logs, and no process/endpoint telemetry** — so it cannot rule out encrypted C2 hiding inside legitimate-looking HTTPS, *traffic to a compromised legitimate site*, or activity that simply wasn't captured. The IP reputation flags also appear unreliable in both directions. For a definitive clearance you would want endpoint logs and timing data. But on the evidence available, the traffic profile is indistinguishable from a normal user browsing the web, and there are **no indications of malware or a cyber-attack**.

NC-26-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this systematically. Let me start by understanding the structure of the Neo4j database, then analyze the traffic for indicators of compromise.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection failed. Let me explore the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me inspect the properties of each node type to understand what data is available.

Used neo4j-database integration

Used neo4j-database integration

Let me understand how the relationships connect these nodes.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have the full picture. Let me identify the local network and characterize traffic. First, the local IPs:

Read Neo4j Cypher

Read Neo4j Cypher

The local network is the 10.0.2.0/24 subnet, with 10.0.2.15 as the single real host (the others are gateway/broadcast/multicast addresses). Let me check flagged external IPs and the conversation volumes.

Read Neo4j Cypher

Read Neo4j Cypher

The graph itself flags 6 external IPs: 2 known attackers and 4 bots. Let me examine the conversations the local host had with each of them.

Read Neo4j Cypher

Read Neo4j Cypher

The local host 10.0.2.15 communicated with **all six** flagged IPs. Let me look at the destination ports, hostnames, and domains for these, and then characterize the host's full activity.

Used neo4j-database integration, ran a command

Used neo4j-database integration, ran a command

That was a large result confirming the same set of external IPs is reached from multiple "vantage points" (the graph models each conversation's perspective). The key finding stands: the local host 10.0.2.15 contacted all six flagged IPs. Let me now characterize the local host's OS and total activity, then check the geographic spread.

Used neo4j-database integration

Used neo4j-database integration

Almost all destination traffic is to standard web ports (443/80); the high-numbered ports are ephemeral source ports. Let me check the conversations with the highest send/receive ratios — exfiltration indicators — and look at the domains the host's bot/attacker IPs belong to in MITRE terms.

Read Neo4j Cypher

Read Neo4j Cypher

One conversation stands out: **77.75.78.150** received ~1 MB uploaded from the host (more upload than download) — unusual for normal web browsing. Let me identify it and the geography of the flagged IPs.

Used neo4j-database integration

Used neo4j-database integration

Let me check what country/city distribution looks like overall and verify the four "yahoo.com" bot IPs more closely, plus check for any beaconing patterns (many small uniform conversations to one host).

Used neo4j-database integration

Used neo4j-database integration

Now let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me also check the MITRE technique for exfiltration over C2 and confirm the threat-intel reputation of the "yahoo.com" hostnames, since those four bot-flagged IPs share an unusual generic hostname.

Searched the web

Searched the web

I have enough to deliver a verdict. Let me do two final checks: the four "yahoo.com" bot IPs (217.12.x and 77.238.x) — these don't actually belong to Yahoo's real address space — and whether the host received data from any flagged IP without initiating (inbound-first contact).

Read Neo4j Cypher

Read Neo4j Cypher

I now have a complete picture. Here is my analysis and verdict.

Verdict: Likely benign — no convincing evidence of malware or cyber-attack

I queried the graph and cross-referenced findings against the MITRE ATT&CK matrix. My assessment is that the captured traffic shows **normal web-browsing activity from a single workstation, with no convincing indicators of compromise.**

What the network looks like

The capture models traffic for one real host on a 10.0.2.0/24 network — 10.0.2.15 (the other "local" addresses are the gateway 10.0.2.2, broadcast 10.0.2.255, multicast 224.0.0.252, and 0.0.0.0/255.255.255.255). This is the default subnet of a VirtualBox NAT VM. The host initiated 198 conversations, downloaded ~75 MB and uploaded ~5 MB, and contacted ~380 external hosts spread across 19 countries (mostly US, Czech Republic, Germany, Ireland, France). Essentially **all** destination traffic went to ports 443 (HTTPS) and 80 (HTTP). That fan-out pattern — hundreds of distinct web servers, download-heavy, all on web ports — is the signature of a person browsing the web, not of malware.

The flagged IPs and why they don't change the verdict

The graph itself flags six external IPs the host talked to: two labeled "known attacker" (37.157.6.253, 216.58.205.238) and four labeled "bot" (217.12.15.83, 77.238.185.35, 217.12.13.41, 217.12.13.40). On the surface this looks alarming, but examining each:

- 216.58.205.238 resolves to fra15s24-in-f14.1e100.net — that is **Google's own infrastructure** (1e100.net is Google). It received the single largest transfer in the capture (7.5 MB down / 0.37 MB sent), which is consistent with normal Google services, not C2.
- 37.157.6.253 resolves to s1.adform.net — **Adform, a mainstream ad-tech/advertising network**. Contact with ad networks is ubiquitous in ordinary browsing.
- The four "bot" IPs (217.12.13.40/41, 217.12.15.83, 77.238.185.35) all carry the placeholder hostname unknown.yahoo.com. Their traffic volumes are small and roughly symmetric (e.g., 16 KB / 5 KB), again typical of web/ad/CDN calls.

These reputation flags appear to be **low-confidence or stale threat-intel labels** applied to shared hosting and ad-network ranges — a very common source of false positives. None of the six exhibits behavior matching a real attack.

MITRE ATT&CK mapping — and why it falls short of a finding

The only technique class plausibly in scope is **T1071.001 – Application Layer Protocol: Web Protocols** (Command & Control) and **T1041 – Exfiltration Over C2 Channel**. Adversaries deliberately use HTTP/HTTPS for C2 because it blends into normal traffic. However, the diagnostic indicators MITRE and detection vendors associate with these techniques are **absent here**:

- **No beaconing.** C2 implants produce many small, periodic, uniform connections to one or a few hosts. This host made only 5 small uniform conversations total, scattered across different destinations — the opposite of a beacon.
- **No exfiltration signature.** T1041 shows as upload volume exceeding download to a suspicious endpoint. Here the host is overwhelmingly download-heavy. The one mildly upload-skewed flow (77.75.78.150, ~1 MB up / 0.65 MB down) goes to a **Prague-hosted server** and is far more consistent with a file/photo upload or sync than data theft.
- **No suspicious destination concentration.** Real C2 funnels to a small number of persistent endpoints. This traffic fans out to hundreds of well-known web/CDN/ad hosts.
- **No non-standard ports or unusual protocols** — everything is 80/443.

Bottom line

The traffic pattern — one workstation, hundreds of web destinations, download-dominant, exclusively on web ports, no periodicity — is characteristic of routine human web browsing. The "attacker"/"bot" reputation flags resolve to Google infrastructure, a legitimate ad network, and generically-labeled shared hosting, and none of them shows attack-consistent behavior. Measured against the MITRE ATT&CK C2/exfiltration techniques, the discriminating indicators are not present.

Conclusion: no indication of malware or cyber-attack on the local network during the capture. Two caveats worth noting: (1) the dataset only contains connection metadata and byte counts — there is no packet payload, TLS-certificate, or timing data, so covert C2 inside HTTPS cannot be 100% excluded; and (2) if you want extra assurance, the 77.75.78.150 upload flow and the four unknown.yahoo.com IPs would be the only items worth a quick second look. Neither rises to the level of an actual finding.

NC-26-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this systematically. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample. Let me explore the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type, starting with the most informative ones.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand the relationship structure between node types.

Read Neo4j Cypher

Read Neo4j Cypher

I have a clear picture of the graph now. Let me identify the local network hosts and analyze the traffic patterns. First, the local/private IPs:

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public (Google's range — 172.217/16 is not RFC1918; only 172.16.0.0/12 is private). The genuine local network is the 10.0.2.0/24 subnet — this is the classic VirtualBox NAT range. Let me characterize the local hosts and their activity.

Read Neo4j Cypher

Read Neo4j Cypher

10.0.2.15 is the host of interest — the VM whose traffic was captured. 10.0.2.2 is the NAT gateway and 10.0.2.255 is the broadcast address. Let me look at the full conversation picture for 10.0.2.15.

Read Neo4j Cypher

Read Neo4j Cypher

One internal host talking to **387 distinct remote peers** in a single capture. Let me look at the remote endpoints — their geography, the domains, and the ports used.

Read Neo4j Cypher

Read Neo4j Cypher

The Czech Republic concentration is notable (47 peers, 20 MB). Let me look at the ports used and the domains contacted.

Read Neo4j Cypher

Read Neo4j Cypher

Traffic is overwhelmingly HTTP/HTTPS with the high-numbered ports being ephemeral client source ports — that's normal. The picture so far is consistent with web browsing. Let me look at the domains contacted and check for any well-known malicious / suspicious indicators.

Read Neo4j Cypher

Read Neo4j Cypher

The domain list looks like ordinary web traffic — Google (1e100.net), Akamai/CloudFront CDNs, AWS, ad networks (adition.com, adnexus.net, turn.com, ioam.de), Czech news/hosting sites (seznam.cz, pszn.cz, casablanca.cz), booking.com, yahoo.com. Now let me query the MITRE ATT&CK matrix to ground my analysis against known adversary techniques, then check for specific indicators.

Searched the web

Searched the web

Now I have the ATT&CK grounding. The key C2 indicators to look for are: **beaconing** (regular, periodic connections), **encrypted channels** masking C2 (T1071.001 — web protocols), traffic to **anomalous/known-bad destinations**, and **data exfiltration** asymmetry. Let me check the byte-volume asymmetry — exfiltration would show unusually high *outbound* (sent-from-local) volume.

Read Neo4j Cypher

Read Neo4j Cypher

The byte ratios are mostly *download-heavy* (ratios 0.02–0.5), the normal signature of web browsing — you send small requests and receive large pages/images/video. The top outbound peer 77.75.78.150 (ratio 1.53, ~1 MB up) is the only mildly upload-skewed one — let me identify it and the other notable peers.

Read Neo4j Cypher

Read Neo4j Cypher

77.75.78.150 is in the 77.75.7x.x range owned by Seznam.cz (the major Czech portal — confirmed by other 77.75.x hosts resolving to seznam.cz). All identified peers are legitimate services. Now the most important ATT&CK check — **beaconing** (T1071). C2 implants connect at regular intervals. Let me check the conversation count distribution and look for repeated

low-volume contact to a single endpoint, plus check for non-standard ports as destination (server-side) ports.

Read Neo4j Cypher

Read Neo4j Cypher

Important finding: the maximum conversation count to any single peer is **2**, and most are **1**. This is the opposite of beaconing. C2 malware produces *many repeated* connections to a *small set* of endpoints over time. Here we see ~387 peers each contacted essentially once — the fan-out signature of a user browsing the web (each site pulls resources from dozens of CDNs/ad networks). Let me confirm there are no destination services on suspicious ports and check the small/empty conversations.

Read Neo4j Cypher

Read Neo4j Cypher

No privileged/server ports other than HTTP and HTTPS are in use at all — no IRC (6667), no SSH, no Telnet, no RDP, no common C2 ports, no DNS-over-nonstandard-port. Let me do two final checks: the broadcast/gateway traffic (10.0.2.2, 10.0.2.255) and verify there's no internal lateral movement.

Read Neo4j Cypher

Read Neo4j Cypher

The only "internal" traffic is a broadcast (likely NetBIOS/SSDP discovery, normal) and gateway traffic from 10.0.2.2 (the NAT router — likely DNS responses). No host-to-host lateral movement, no internal scanning. Let me do one consolidated check of the conversations involving unidentified-port (null) traffic, since those weren't HTTP/HTTPS.

Read Neo4j Cypher

Read Neo4j Cypher

The null-port traffic is all benign: limited broadcasts (255.255.255.255, 10.0.2.255), an LLMNR multicast (224.0.0.252), gateway DNS (10.0.2.2), and Fortinet DNS/services (208.91.112.52/53 — fortinet.com). Nothing anomalous.

I have everything I need for a verdict. Let me give you the full analysis.

Verdict: No indications of malware or cyber-attack

Summary

After querying the captured-traffic graph and grounding the analysis against the MITRE ATT&CK framework, **the traffic from the local network is consistent with ordinary, benign web browsing by a single user**. No technique from the ATT&CK matrix is evidenced in the data.

What the graph contains

The capture models one active local host on a 10.0.2.0/24 subnet (the VirtualBox NAT default range):

- **10.0.2.15** — the monitored endpoint, party to all 394 conversations
- **10.0.2.2** — the NAT gateway; **10.0.2.255** — the subnet broadcast address

This host exchanged traffic with **387 distinct external peers** (~34 MB sent / ~78 MB received).

ATT&CK-based analysis

I checked the data against the adversary behaviors the framework says network traffic can reveal — primarily Command & Control (TA0011) and Exfiltration (TA0010), since those are what netflow-style data can show:

Command & Control / Beaconing (T1071) — Not present. C2 implants connect *repeatedly* to a *small set* of endpoints at regular intervals. The data shows the exact opposite: the maximum conversation count to *any* single peer is **2**, and the overwhelming majority of peers were contacted **once**. A wide fan-out of one-shot connections is the signature of a browser loading pages — each website pulls resources from dozens of CDNs and ad networks.

Application-Layer Protocol abuse (T1071.001) — No evidence. Traffic is ~99% HTTP (88 conversations) and HTTPS (328). While malware *can* hide C2 in HTTPS, here the destinations and traffic shape rule it out (see below). All non-HTTP "ports" resolve to standard broadcast/multicast discovery (LLMNR, NetBIOS, DHCP) and DNS — all expected.

Non-standard ports / unusual services — None. No privileged destination port other than 80/443 appears anywhere. No IRC, SSH, RDP, Telnet, or known C2 ports. The thousands of high-numbered ports are ephemeral client *source* ports — normal TCP behavior.

Exfiltration (TA0010) — No evidence. Byte ratios are strongly *download-heavy* (typically 0.02–0.5 sent/received), the normal browsing pattern. The single mildly upload-skewed peer (77.75.78.150, ratio 1.53, ~1 MB up) belongs to **Seznam.cz**, the Czech web portal — consistent with form posts/uploads to a legitimate service.

Lateral Movement (TA0008) / Discovery — None. The only intra-subnet traffic is one broadcast and gateway DNS. No host-to-host connections, no internal scanning.

Destination reputation

Every identifiable endpoint is a well-known legitimate service: Google (1e100.net), Akamai and CloudFront CDNs, AWS, Microsoft (a-msedge.net, Bing), Yahoo, booking.com, fortinet.com, mainstream ad networks (adnexus.net, adition.com, turn.com, ioam.de), and Czech/Slovak sites (seznam.cz, pszn.cz, casablanca.cz, cez.cz, webglobe.sk). The geographic spread (US, Czech Republic, Germany, Ireland, France) matches CDN edge distribution plus a Czech-based user. No known-malicious domains or IPs, no algorithmically-generated domains, no bulletproof-hosting ranges.

Caveats on the verdict

This is a **clean assessment for the data available**, but the graph is netflow-level, so a few blind spots remain by construction: there are **no timestamps**, so I cannot rule out *slow* beaconing definitively by timing (though the once-per-peer pattern argues strongly against it); there is **no payload/JA3/TLS-certificate data**, so well-disguised HTTPS C2 to a legitimate-looking domain (domain fronting, T1090.004) cannot be excluded by traffic shape alone; and the capture is a single bounded window. Within those limits, nothing in the graph points to compromise.

Bottom line: the capture shows one VM browsing the web normally. No malware, no C2, no exfiltration, no lateral movement — no attack indicators.

If you'd like, I can produce this as a formal written incident report, or — if the source capture has timestamps that weren't loaded into the graph — a timing-based beaconing analysis would close the main remaining blind spot.

NC-26-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate the Neo4j database for indications of malicious network activity. Let me start by loading the necessary tools and examining the database schema.

Searched available tools

Searched available tools

Let me examine the database schema first.

Get Neo4j Schema

Get Neo4j Schema

The default schema call has an issue. Let me query the schema manually.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models: IPs, Hosts, OS, Ports, Conversations, and geolocation (City/Country/Domain). Let me explore the data to understand the network.

Read Neo4j Cypher

Read Neo4j Cypher

The local host appears to be 10.0.2.15 (a standard VirtualBox NAT address — this is a single-host capture). Let me identify all local-network IPs and look at the broader picture.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are public (Google ranges), not private. The true local network is the VirtualBox NAT subnet 10.0.2.0/24: host 10.0.2.15, gateway 10.0.2.2, broadcast 10.0.2.255. So this is a single-machine capture. Let me look at the host details and ports used.

Used neo4j-database integration

Used neo4j-database integration

The host 10.0.2.15 uses an enormous range of sequential ephemeral source ports — over 2,000 of them. Let me quantify the distinct destinations and check the overall conversation structure.

Used neo4j-database integration

Used neo4j-database integration

Interesting — the destination services are almost entirely HTTPS (443) and HTTP (80), normal web traffic. Now let me look at the geographic distribution and the domains involved, then examine the receiver-side ports (the actual server ports the host connected to) to find anything unusual.

Used neo4j-database integration

Used neo4j-database integration

All server-side traffic is on standard web ports (443 HTTPS and 80 HTTP). No non-standard listening ports, no IRC, no high-port C2 channels. Let me examine the geographic spread and the domains, and check for any beaconing-like patterns or suspicious destinations.

Used neo4j-database integration

Used neo4j-database integration

The domain profile is entirely benign-looking — Google (1e100.net), AWS, Akamai, CloudFront, ad networks, Czech news/shopping sites (seznam.cz, zbozi.cz), Booking.com, Fortinet. This looks like ordinary web browsing from a machine in the Czech Republic.

Now let me check for the patterns that *would* indicate compromise: data exfiltration (high upload-to-download ratios) and beaconing. Let me look at conversations where the host sent far more than it received.

Read Neo4j Cypher

Read Neo4j Cypher

Only two conversations show the host uploading more than downloading, and both at modest ratios (~1.3–1.5×) with destinations that are CDN/Facebook infrastructure (31.13.92.x is Facebook, 77.75.x is Seznam.cz). No large-scale exfiltration pattern. Let me check the receiver-side ports per remote IP to confirm there is no non-standard service, and look at the per-conversation port spread to assess beaconing.

Read Neo4j Cypher

Read Neo4j Cypher

Confirmed: every server-side port in the capture is either 443, 80, or an ephemeral high port. No backdoor listeners, no unusual services. Now let me look at the MITRE ATT&CK matrix to ground the verdict against a recognized framework. Let me check the current version and the relevant network-observable techniques.

Searched the web

Searched the web

I have the current MITRE ATT&CK context (Enterprise v19, 15 tactics). Let me do a couple of final targeted checks on the graph — looking for any IP that received an unusually high number of distinct connections (a beaconing/C2 signature) and any conversation with a very small, regular payload size.

Read Neo4j Cypher

Read Neo4j Cypher

The IPs with many connections are all high-volume CDN/cloud endpoints (Akamai 23.63.x / 23.51.x, Google 216.58.x, CERN 188.184.x, Edgecast 93.184.220.x) carrying megabytes of data — that is the signature of a browser loading rich web content, not low-and-slow C2 beaconing (which would show many connections carrying tiny, near-identical payloads). Let me do one last check: small-payload conversations that could be beacons.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a verdict. Let me give you a visual summary alongside the analysis.

Based on my investigation of the graph database and the MITRE ATT&CK framework, here is a visual summary followed by my verdict.

Verdict: No indications of malware or cyber-attack

I queried the Neo4j graph and assessed it against the MITRE ATT&CK Enterprise matrix (v19, the current release). The capture shows no evidence of compromise.

What the graph actually contains

The "local network" is a single machine. The only RFC 1918 private address in the graph is 10.0.2.15, sitting on the 10.0.2.0/24 VirtualBox NAT subnet (with 10.0.2.2 as gateway and 10.0.2.255 as broadcast). Every other address is a public Internet host. So this is a host-level capture of one workstation's outbound activity over the capture window — roughly 113 MB across 395 conversations to 388 distinct remote IPs.

Why this reads as benign, mapped to ATT&CK

The reasoning is partly about what's present and partly about what's conspicuously absent:

Command & Control (T1071, T1571). Every single server-side port in the capture is 443 (HTTPS, 328 conversations) or 80 (HTTP, 88 conversations) — there are zero non-web listening ports. The high-numbered ports (49000–51000) are sequential ephemeral client source ports, which is exactly how an OS allocates outbound TCP sockets for a browser. There's no IRC, no odd high-port service, no protocol-on-wrong-port. The destinations are mainstream infrastructure: Google (1e100.net), AWS, Akamai, CloudFront, ad networks, Fortinet, plus Czech consumer sites (seznam.cz, zbozi.cz, booking.com). No domains resembling C2 or DGA infrastructure.

Exfiltration (T1041, T1048). Only 2 of 395 conversations had the host sending more than it received, and both at modest ratios (~1.3–1.5×) to CDN/Facebook endpoints — normal for form posts and uploads. The traffic is overwhelmingly inbound (78 MB down vs 34 MB up), the signature of content consumption, not data theft.

Beaconing. The high-connection-count IPs (Akamai, Google, CERN, Edgecast) each carry megabytes of varied payload — that's a browser pulling page assets. Low-and-slow C2 beaconing would instead show many connections with tiny, near-uniform payloads to a single endpoint; that pattern is not present.

Discovery & Lateral Movement (T1046, TA0008). With only one host on the local subnet, there is no internal scanning, no host-to-host traffic, nothing to laterally move toward.

Geographically, traffic concentrates in the US, Czech Republic, Germany, and Ireland — consistent with a Czech-based user browsing the web behind major CDNs.

Caveats on scope

This conclusion is bounded by what the graph models. The dataset has no timestamps, no packet-level detail, no DNS query records, no TLS certificate or JA3 data, and no payload inspection. A capture like this cannot rule out a well-disguised threat that communicates purely over HTTPS to a reputable-looking host (e.g. malware abusing a cloud service for C2). It also can't detect host-side activity that produces no network traffic. So the accurate statement is: within the connection-level metadata available, there are no indicators of compromise — not an absolute guarantee of a clean machine.

If you'd like to go deeper, the useful next steps would be adding timestamp data to test for beacon periodicity, and DNS/TLS metadata to vet destination reputation properly.

NC-26-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this network traffic graph for signs of malicious activity. Let me start by loading the Neo4j tools and examining the database structure.

Searched available tools

Searched available tools

Let me examine the database schema first.

Get Neo4j Schema

Get Neo4j Schema

The default sample parameter is causing an issue. Let me specify a sample size explicitly.

Get Neo4j Schema

Get Neo4j Schema

The graph models a network capture with IPs, Hosts, Ports, Conversations, and geolocation. Let me explore the data in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the local network hosts, the ports in use, and the geographic distribution of remote endpoints.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are public Google ranges, not RFC1918. The actual local network is 10.0.2.0/24 — a typical VirtualBox NAT subnet, with 10.0.2.15 being the host of interest, 10.0.2.2 the gateway, and 10.0.2.255 broadcast. Let me examine this host and its connections.

Used neo4j-database integration

Used neo4j-database integration

That's the source-side ephemeral port range — normal for an active client. The diagnostic information is in the destination ports of conversations. Let me look at what the local host actually connected to.

Read Neo4j Cypher

Read Neo4j Cypher

Most traffic is HTTPS/HTTP to mainstream CDNs and services. Now let me look at the geographic distribution of remote endpoints and check for any unusual destination ports beyond 80/443.

Read Neo4j Cypher

Read Neo4j Cypher

So the only well-known service ports in the entire capture are 80 (HTTP) and 443 (HTTPS). Everything else is ephemeral client ports. There's a null port conversation worth checking. Let me examine that, plus the geographic spread and the per-country endpoint breakdown.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the null port conversation and look at the domains observed in the capture.

Used neo4j-database integration

Used neo4j-database integration

Now I have a good picture of the dataset. Let me query the MITRE ATT&CK framework to ground my analysis in the standard taxonomy, focusing on network-observable techniques.

Searched the web

Searched the web

Let me run a few targeted Cypher checks for the specific network-observable indicators — beaconing patterns (many small repeated connections), data-volume mismatches that could indicate exfiltration, and connections to suspicious destinations.

Read Neo4j Cypher

Read Neo4j Cypher

No outbound-heavy "exfiltration" pattern — the most outbound-skewed real conversation (77.75.78.150, a Seznam.cz IP) is only 1.5:1 and modest. Let me check connection counts per remote IP (beaconing indicator) and look at the breadth of distinct destinations.

Read Neo4j Cypher

Read Neo4j Cypher

394 conversations with 387 distinct remote endpoints — essentially a 1:1 ratio. That's the signature of normal web browsing (each site/CDN contacted once), **not** beaconing, which would show many repeated connections to a *small* number of endpoints. Let me do a final check on the highest-volume conversation and verify the overall traffic direction.

Read Neo4j Cypher

Read Neo4j Cypher

I have everything I need. Here's my verdict.

Verdict: No indications of malware or cyber-attack

After querying the graph against network-observable MITRE ATT&CK techniques, the capture is consistent with **ordinary web browsing by a single host**. There are no indicators of compromise.

What the graph contains

The capture models one active local host, **10.0.2.15** (a VirtualBox NAT subnet, 10.0.2.0/24, with 10.0.2.2 as gateway and 10.0.2.255 broadcast). It held **394 conversations** with **387 distinct remote endpoints** across 19 countries — Google, Facebook (31.13.92.x), Amazon AWS/CloudFront, Akamai, Microsoft (a-msedge.net), and Czech services (seznam.cz, booking.com, billboard.cz). Total traffic was ~8 MB out / ~104 MB in.

Why the evidence points to benign activity

No command-and-control beaconing (T1071/T1102). Beaconing means many repeated, regular check-ins to a *few* endpoints. Here the ratio of conversations to distinct destinations is ~1:1 — almost every remote IP is contacted exactly once. That is the fingerprint of a browser loading many different sites and CDNs, the opposite of C2.

No data exfiltration (T1041/T1048). Exfiltration shows an outbound-heavy volume mismatch. This host's overall ratio is **~13:1 inbound** (downloading), and even the single most outbound-skewed conversation is only 1.5:1 with modest volume. MITRE's detection strategy for exfiltration looks for suspicious outbound traffic volume mismatches — nothing of the sort exists here. MITRE

No suspicious ports or protocols. Every destination port in the entire capture is **80 (HTTP)** or **443 (HTTPS)** — 88 conversations on HTTP, the rest HTTPS. All other ports are normal client-side ephemeral ports (49000–51000). There are no non-standard listening ports, no IRC/SSH/RDP, no signs of lateral movement or remote-access tooling.

Destinations and domains are reputable. Resolved domains are mainstream CDNs, cloud providers, ad networks, and well-known consumer sites. No newly-registered, algorithmically-generated (DGA), or known-malicious domains appear.

The one quirk is benign. The conversation to 208.91.112.53 with a null port (2 MB exchanged) lacks port metadata — likely a parsing gap in the capture, not malice. 208.91.112.53 belongs to Fortinet/Sophos-style infrastructure and the volume/direction are unremarkable.

Caveats

This is a flow-level graph: it has byte counts, IPs, ports, and geolocation, but **no timestamps, packet payloads, TLS certificate details, or JA3 fingerprints**. So it cannot rule out a very stealthy, low-and-slow C2 channel that perfectly mimics HTTPS to a legitimate-looking host — exfiltrating over an existing C2 channel is stealthy precisely because it creates no new outbound connections and blends with normal traffic. But nothing in the available data *suggests* that; a verdict can only be rendered on observable evidence, and the observable evidence is clean. Startupdefense

Bottom line: the capture shows a single workstation doing normal HTTPS web browsing. No malware, no C2, no exfiltration, no lateral movement.

NC-28-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this for you. Let me start by exploring the Neo4j database to understand the network traffic model, then cross-reference findings against the MITRE ATT&CK framework.

Searched available tools

Searched available tools

Let me first inspect the database schema to understand how the network traffic is modeled.

Get Neo4j Schema

Get Neo4j Schema

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

The schema is clear. Let me map out the data model and explore the key entities.

Graph model:

- IP nodes (712) — with flags is_Local, is_spam, is_known_attacker, is_Bot
- CONVERSATION nodes (716) — with sender, receiver, bytes_sent, bytes_received
- PORT nodes (4332), Host, OS, Domain, City, Country

Let me start gathering the security-relevant signal. First, the flagged IPs:

Read Neo4j Cypher

Read Neo4j Cypher

Seven local IPs, and among external IPs there are 7 known attackers and 11 bots. Let me identify the local hosts and the flagged external IPs.

Read Neo4j Cypher

Read Neo4j Cypher

The local subnet is 10.0.2.0/24 (plus a 192.168.33.254 gateway). The actual host is **10.0.2.15** — the others are infrastructure addresses (0.0.0.0, .2 gateway, .255 broadcast, 224.0.0.252 LLMNR multicast, 255.255.255.255 broadcast). Let me examine the flagged external IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the conversations involving the local host and these flagged IPs. First, the conversations the local host had:

Read Neo4j Cypher

Read Neo4j Cypher

Let me focus on the conversations specifically with the flagged attacker/bot IPs, and also pull the ports used in those conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a detailed picture. Let me check the domains and ports involved more broadly, and also look at the overall port usage to spot any non-standard service ports.

Used neo4j-database integration

Used neo4j-database integration

Good — only standard service ports appear (443, 80, 53). No anomalous listening services. Let me check the domains the local host's hostname is associated with and the domains in the dataset, then look at whether the host node exists for the local IP.

Read Neo4j Cypher

Read Neo4j Cypher

The flagged "attacker" IPs resolve to well-known infrastructure: Twitter (199.16.156.x / 104.244.46.7), Facebook (31.13.84.4), Akamai (63.215.202.137). The "bot" IPs are Yandex/Mail.ru ranges (87.250.x, 93.158.134.x, 213.180.193.x) and 87.248.118.x (Yahoo). This is important context. Let me check one suspicious one — 203.130.61.92 (Beijing), which had a notably large download and many sequential ephemeral ports — and look at its domain.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me verify the MITRE ATT&CK framework against these observations. Let me query the official ATT&CK source for the relevant techniques.

Searched the web

Searched the web

I have enough to assess against ATT&CK. The " unknown" ports in the data are simply ephemeral client-side source ports (49000–61000 range) — that's normal TCP behavior, not T1571 (which concerns *server-side* non-standard ports). Let me do a couple of final checks: the conversation count per local-to-flagged pair and any beaconing-style pattern, plus whether any local IP received inbound connections that look like a listening service.

Read Neo4j Cypher

Read Neo4j Cypher

No inbound connections to SSH/RDP/SMB/Telnet/RPC on the local host — no signs of remote-service exploitation or a backdoor listener. Let me do one last check: total data volume direction for the local host (to spot exfiltration) and the IP→Country spread of its conversations.

Read Neo4j Cypher

Read Neo4j Cypher

I have everything needed for a verdict. Here is my assessment.

Verdict: No reliable indication of malware or cyber-attack against the local network

What the graph contains

The capture models a single active local host — **10.0.2.15** (a 10.0.2.0/24 NAT subnet, the rest of the "local" IPs are infrastructure addresses: gateway, broadcast, LLMNR multicast). It had **715 conversations** with 705 external IPs, downloading ~128 MB and uploading ~10 MB.

Findings cross-referenced against MITRE ATT&CK

1. The "ephemeral ports" are not Non-Standard Port (T1571). The thousands of high-numbered ports (49000–61000) flagged "unknown" in the graph are **client-side ephemeral source ports** — normal, OS-assigned TCP behavior for every outbound connection. ATT&CK T1571 concerns a *service* running on an unexpected *destination* port (e.g., HTTPS on 8088). Every actual service port in the dataset is standard: **443, 80, and 53 only**. This is the single most important point — it's the most likely thing to be misread as malicious, and it isn't.

2. The "known attacker" IPs are mislabeled mainstream infrastructure. The 7 IPs flagged is_known_attacker resolve to major platforms: 199.16.156.73/.201 and 104.244.46.7 are **Twitter**, 31.13.84.4 is **Facebook**, 63.215.202.137 is **Akamai**, 205.201.132.96 is a **mail/marketing provider**. All traffic to them was ordinary HTTPS web browsing. These flags appear to be stale or low-quality threat-intel labels, not evidence.

3. The "bot" IPs are search-engine/webmail infrastructure. The 11 is_Bot IPs are **Yandex** (87.250.x, 93.158.134.119, 213.180.193.119), **Mail.ru**, and **Yahoo** (87.248.118.x) ranges. The large 7 MB transfer from 87.248.118.22 is a Yahoo content download — consistent with normal webmail/portal use, not C2.

4. No Command-and-Control signature. C2 (TA0011) typically shows small, regular, symmetric beacons to a stable endpoint. Here the traffic is bursty, heavily **download-skewed (upload ratio 0.078)**, and spread across 705 destinations — the profile of a person browsing the web, not a beaconing implant.

5. No Exfiltration (TA0010). Exfil shows an abnormal *upload* skew to few destinations. The opposite is true here: ~13× more downloaded than uploaded.

6. No inbound exploitation or backdoor. Zero inbound connections to SSH/RDP/SMB/Telnet/RPC on 10.0.2.15. No evidence of External Remote Services abuse (T1133), Exploitation of Remote Services (T1210), or a listening backdoor.

The one item worth a footnote

203.130.61.92 (Beijing, reverse DNS 203.130.61.92-BJ-CNC) received a 3.4 MB download from the host. The pattern (HTTPS, download-skewed, sequential ephemeral source ports) is still consistent with content download rather than C2, but because it's a bare-IP host in a Chinese ISP block with no clean domain, it's the only endpoint I'd suggest a human analyst eyeball if raw packets are available.

Bottom line

The dataset's own is_known_attacker / is_Bot flags are unreliable (they tag Twitter, Facebook, Yahoo and Yandex), and the high port numbers are a benign artifact of how TCP works. Mapped against ATT&CK, **none of the C2, Exfiltration, Lateral Movement, or Remote-Service tactics show a genuine match.** The capture looks like normal outbound web and webmail activity from one workstation. **No credible indication of malware or cyber-attack** — with the minor caveat that this graph lacks timestamps, packet payloads, and process attribution, so it cannot *fully* rule out a low-and-slow compromise; that would need the original PCAP.

NC-28-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate the network traffic graph for signs of malicious activity. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection failed. Let me explore the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type and how relationships connect them.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand how the relationships connect the nodes.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a clear picture of the graph structure. Let me identify the local network hosts and analyze their traffic.

Read Neo4j Cypher

Read Neo4j Cypher

The local network has one real host (10.0.2.15) plus infrastructure addresses (gateway 10.0.2.2, broadcast, multicast). Let me check for flagged external IPs and analyze the traffic patterns.

Read Neo4j Cypher

Read Neo4j Cypher

There are 18 flagged external IPs (11 bots, 7 known attackers). Let me see what conversations the local host had with these, and resolve their hostnames.

Read Neo4j Cypher

Read Neo4j Cypher

Important context: many "bot"/"attacker" flagged IPs here resolve to legitimate ad/CDN/analytics infrastructure (Yandex mc.yandex.ru, bs.yandex.ru, Yahoo, Facebook CDN,

Dotomi ad network, Twitter 199.16.156.x). These flags often come from threat-intel feeds that classify ad-tracking and crawler infrastructure as "bots." Two items stand out as genuinely anomalous in volume, so let me investigate them and the overall traffic profile.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the ports used by the local host, especially with the flagged IPs, and check for unusual destination ports.

Ran a command, used neo4j-database integration

Ran a command, used neo4j-database integration

These are ephemeral client source ports (49xxx range, each used once — normal). Let me look at the *destination* ports — i.e. server-side ports — to see what services the local host actually connected to.

Used neo4j-database integration

Used neo4j-database integration

All non-ephemeral destination ports are 80/443 — standard web traffic only. No backdoor/C2/IRC/RDP/SMB ports. Now let me examine the two most anomalous conversations more closely: the China-based attacker IP (large download) and the highest-volume conversation overall.

Used neo4j-database integration

Used neo4j-database integration

Now let me get the overall traffic statistics and check the direction of data flow (outbound vs inbound) to assess any exfiltration pattern.

Used neo4j-database integration

Used neo4j-database integration

Now I have the full picture. When the local host initiates (sender), it sends 5 MB and receives 77 MB — normal web browsing (download-heavy). When it's the receiver, the remote sends 50 MB. So net inbound dominates — consistent with content consumption, not exfiltration. Let me check how many distinct external destinations and countries, and look for any beaconing pattern (many tiny conversations to one host).

Used neo4j-database integration

Used neo4j-database integration

No beaconing pattern — virtually every external IP has just one conversation. Let me check the domains contacted for any DGA-like (algorithmically-generated) or suspicious names, and check the OS of the local host.

Read Neo4j Cypher

Read Neo4j Cypher

Two hostnames stand out: *.darkishaeromedical.com and *.scatsideview.net — these have the random-word, low-reputation appearance typical of malicious or compromised infrastructure. Let me examine those conversations in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Notable: all three suspicious-named hosts are in the same 185.42.119.0/24 block (France). The 185.42.119.0/24 range is well-documented as belonging to malicious/exploit-kit hosting. Let me check what other IPs in that subnet were contacted.

Read Neo4j Cypher

Read Neo4j Cypher

Let me also check the domain-level data and look at the HTTP (port 80) conversations specifically, as those reveal more than encrypted ones. First, let me query the MITRE ATT&CK matrix to map these findings to documented techniques.

Searched the web

Searched the web

The MITRE T1189 detection guidance is directly relevant. Let me do a couple more targeted queries to firm up the verdict — checking the HTTP conversations and any conversations to Russia/China with significant volume.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a thorough verdict. Let me compile the analysis.

Verdict: Network Traffic Threat Assessment

Summary

There are weak-to-moderate indications of suspicious activity, but no confirmed evidence of an active malware infection or successful cyber-attack during the capture window. The traffic profile is overwhelmingly consistent with ordinary web browsing. The findings below warrant follow-up but do not, on their own, constitute proof of compromise.

Scope of the captured graph

The database models one capture session: 716 conversations, 712 IPs, 130 domains across 23 countries. The local network contains a **single active host — 10.0.2.15** (a NAT'd 10.0.2.0/24 address, almost certainly a VirtualBox VM) plus its gateway (10.0.2.2), broadcast and multicast addresses. So this is effectively a single-endpoint capture, not a multi-machine LAN.

What looks normal (the majority of evidence)

- **Ports:** Every non-ephemeral destination port is **80 (HTTP) or 443 (HTTPS)**. There are no connections to backdoor/C2-associated ports (no RDP, SMB, IRC, SSH, or high

non-standard listeners). The thousands of 49xxx ports are normal ephemeral client source ports, each used once.

- **No beaconing:** 701 of 705 external IPs were contacted exactly once; the rest twice. There is no periodic call-home pattern to a single host — a hallmark **absence** of typical C2 (ATT&CK T1071 / T1571).
- **Data direction:** When 10.0.2.15 initiates a connection it sends ~5 MB and receives ~77 MB — a download-heavy ratio typical of content consumption, **not** exfiltration. There is no large asymmetric upload to a single odd destination.
- **Flagged "bot"/"attacker" IPs are largely false positives:** Most flagged IPs resolve to legitimate ad-tech, CDN, and analytics infrastructure — mc.yandex.ru, bs.yandex.ru, avatars.mds.yandex.net, Yahoo, fbcdn.net (Facebook CDN), and dotomi.com (an ad network). Threat-intel feeds routinely tag advertising/crawler infrastructure as "bots." The large 87.248.118.22 "upload" (~7 MB) goes to a Yahoo address over 443 and is consistent with normal web-app traffic.

What is genuinely suspicious (the findings to escalate)

1. Low-reputation, randomly-named domains in a known-bad subnet. The local host contacted three IPs in **185.42.119.0/24** (France), all over HTTPS:

IP	Host	Bytes (sent / recv)
185.42.119.33	*.scatsideview.net	31,505 / 8,966
185.42.119.65	*.scatsideview.net	1,650 / 5,352
185.42.119.129	*.darkishaeromedical.com	176 / 400

The hostnames (scatsideview.net, darkishaeromedical.com) have the random-dictionary-word character associated with **exploit-kit landing pages and malvertising redirectors**. MITRE's detection guidance for **Drive-by Compromise (T1189)** specifically calls out *network requests to previously unseen/suspicious domains* as a primary indicator. This is the single most concerning finding.

2. A China-based host with a large inbound transfer. 203.130.61.92 (hostname 203.130.61.92-BJ-CNC, China Unicom Beijing, also flagged is_known_attacker) sent **~3.4 MB to the local host over HTTP**. A multi-megabyte HTTP download from a flagged Chinese IP is worth verifying — though it is equally consistent with a large file or media download.

Mapping to MITRE ATT&CK

- **T1189 – Drive-by Compromise:** The contact with low-reputation, randomly-named domains is the classic *initial-access* indicator. However, confirming this technique requires endpoint evidence MITRE lists — browser spawning unusual child processes, script-engine execution, file drops — **none of which exist in this graph** (it has no process or host-execution data).
- **T1071 – Application Layer Protocol / T1571 – Non-Standard Port (C2): Not supported.** No beaconing and no non-web ports.

- **T1041 – Exfiltration over C2: Not supported.** Net flow is strongly inbound.

Why this is a "weak indication" and not a confirmed attack

The graph captures only **network-level metadata** (IPs, ports, byte counts, DNS). It contains no payloads, no HTTP URIs/headers, no process or filesystem data. The suspicious-domain contact is enough to flag the endpoint for investigation, but a visit to a malicious-looking domain is not the same as a successful exploitation. The defining evidence of compromise — exploit delivery, code execution, persistence, C2 callbacks — is **absent**, and that absence is meaningful: a successfully infected host would typically show beaconing or exfiltration, and this one does not.

Recommendation

Treat 10.0.2.15 as **suspect, not confirmed-infected.** Specifically:

1. Investigate the 185.42.119.0/24 HTTPS sessions — pull full PCAP/proxy logs for those flows if available, and check the host for post-visit indicators.
2. Review the 203.130.61.92 3.4 MB HTTP download — identify what was transferred.
3. Run an endpoint scan and review browser/process/persistence artifacts on 10.0.2.15, since the graph cannot answer the question of whether a drive-by actually succeeded.

If you can load additional data (HTTP request URIs, file hashes, or host process events) into Neo4j, I can extend the analysis to confirm or rule out T1189 definitively.

NC-28-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC isn't cooperating. Let me inspect the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me look at sample nodes for each label to understand the properties.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand the relationship structure.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me find the local network hosts — the actual subjects of the investigation.

Read Neo4j Cypher

Read Neo4j Cypher

The local network has one real host: 10.0.2.15 (the others are gateway, broadcast, multicast addresses). Let me check the flagged remote IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Several flagged remote IPs. Let me see which of these the local host actually communicated with, and the conversation volumes.

Read Neo4j Cypher

Read Neo4j Cypher

Key findings so far. The local host 10.0.2.15 communicated with all 18 flagged IPs (10 "bots", 8 "known attackers"). Notable:

- 203.130.61.92 (known attacker): host **received 3.39 MB** — a large download.

- 87.248.118.22 (bot): host **sent 7.08 MB** — a large upload.
- 104.244.46.7 (known attacker): host sent 284 KB.

Let me look at the ports and the destination identities (domains/hosts) to understand the nature of these connections.

Read Neo4j Cypher

Read Neo4j Cypher

This is very revealing. The "flagged" hosts resolve to legitimate, well-known infrastructure. Let me get the full picture of all destinations, then query the MITRE ATT&CK matrix for context. First, let me look at what services the local host contacted overall.

Used neo4j-database integration

Used neo4j-database integration

Good — only standard service ports used (HTTPS, HTTP, DNS); all the high-numbered ports are normal client ephemeral ports. Now let me check conversation symmetry/byte patterns for the local host, and whether any peer count is unusually high (beaconing/scanning).

Used neo4j-database integration

Used neo4j-database integration

Interesting — the host "uploads" a lot to many CDN/cloud destinations. This is characteristic of a packet capture where bytes_sent/received labelling may be reversed, or simply heavy web traffic. Let me check the geographic spread and domains contacted to characterize the overall traffic profile.

Used neo4j-database integration

Used neo4j-database integration

I have a clear picture of the graph. Now let me query the MITRE ATT&CK matrix as instructed, to map these observations against known adversary techniques.

Searched the web

Searched the web

Let me fetch the authoritative MITRE tactics list and then map specifically the network-observable techniques (C2, exfiltration) which is what a traffic capture can actually reveal.

Searched the web

Searched the web

I have enough to map the evidence against MITRE ATT&CK. Let me run a couple more targeted queries to check for the classic network-observable attack signatures — beaconing patterns and unusual byte ratios.

Read Neo4j Cypher

Read Neo4j Cypher

Note the conversation IDs are all <remote>-10.0.2.15, where remote is listed as the "sender." In this dataset, the "sender" is the host that initiated/sourced the bulk of the data — these are servers delivering content (downloads, video, web pages) to the local host. The byte directions are consistent with a workstation browsing the web, not exfiltration. Let me do one final check — the DNS conversations and whether any single remote host shows beacon-like repeated small conversations.

Read Neo4j Cypher

Read Neo4j Cypher

No peer has more than 2 conversations with the host — there is no repetitive beaconing pattern. Here is my analysis.

Verdict: No credible indication of malware or cyber-attack

After mapping the captured traffic of the only real local host (10.0.2.15) against the MITRE ATT&CK Enterprise matrix, the evidence does **not** support a finding of compromise. The capture is consistent with a single workstation doing ordinary web browsing.

What the graph contains

The local network has one active host, 10.0.2.15 (the other "local" IPs — 10.0.2.2, 10.0.2.255, 224.0.0.252, 255.255.255.255, 0.0.0.0 — are a gateway, broadcast and multicast addresses). It held 715 conversations with 710 distinct external peers, almost entirely over ports **443 (HTTPS)**, **80 (HTTP)** and **53 (DNS)**. The peer set is dominated by mainstream infrastructure: Akamai, CloudFront, Amazon AWS, Google (1e100.net), Facebook/fbcdn.net, Yandex/Rambler, dailymotion, and ad networks.

Why the "flagged" IPs are not convincing evidence

The database flags 18 remote IPs as is_known_attacker or is_Bot. On inspection these reverse-resolve to **legitimate, well-known services**, not adversary infrastructure:

- "Known attackers" 199.16.156.73/.201 and 104.244.46.7 are Twitter/X ranges; 31.13.84.4 is fbcdn.net (Facebook CDN); 63.215.202.137 is dotomi.com (an ad network); 203.130.61.92 resolves to a Chinese ISP host serving normal HTTP/HTTPS content.
- The "bots" are Yahoo (unknown.yahoo.com) and Yandex (mc.yandex.ru, bs.yandex.ru, avatars.mds.yandex.net) servers — analytics and content hosts, the kind of third-party calls any web page triggers.

These look like generic reputation/geolocation tags (likely "non-US" or "ad-related") rather than verified threat intelligence. Contacting them once each, over 443, is expected behavior.

Mapping to MITRE ATT&CK — what a traffic capture *could* show, and what's absent

A network capture can realistically reveal only the **Command and Control** (TA0011) and **Exfiltration** (TA0010) tactics, plus network-side **Discovery/Reconnaissance**. Checking each:

- **C2 — Application Layer Protocol (T1071)** / non-standard ports (T1571): Not seen. All traffic uses standard service ports; no odd ports, no plaintext on unusual ports.

- **C2 beaconing**: Not seen. No remote peer was contacted more than twice — there is no periodic, repetitive callback pattern, which is the hallmark of an implant. 710 one-off destinations is a browsing fan-out, the opposite of beaconing.
- **Exfiltration Over C2 Channel (T1041)** / **to Cloud (T1567)**: Not supported. Yes, several conversations are "send"-heavy (e.g. 7 MB, 5 MB), but in this dataset the conversation id is <source>-<destination>, and those high-volume flows are sourced *by the remote server* (CloudFront, Akamai, Fastly 151.101.x, Google) — i.e. **content being delivered to the host** (downloads, video, images), not data leaving it. The destinations are CDNs, not collection points, and each is a single transfer.
- **Reconnaissance / Active Scanning (T1595)**: Not seen. No internal scanning, no sequential port sweeps, no lateral movement (only one host present anyway).

Tactics like Initial Access, Execution, Persistence, Privilege Escalation, Credential Access and Impact are **endpoint behaviors** this capture cannot observe — so their absence here is not proof of safety, just outside the data's scope.

Bottom line

The traffic profile — one host, hundreds of one-shot HTTPS/HTTP/DNS connections to CDNs, cloud providers, social media and ad/analytics networks, with no beaconing, no anomalous ports, and no genuine outbound exfiltration pattern — matches **normal web browsing**. The IPs the database marks as "attacker"/"bot" are mislabeled mainstream services and do not change that assessment.

Confidence: moderate-to-high that there is no attack visible *in this capture*. Two caveats: (1) the dataset has no timestamps, so I could not verify timing-based beaconing precisely; and (2) a network capture cannot rule out host-level compromise that produced no distinctive network traffic. If you want deeper assurance, endpoint logs (process/auth events) would be the place to look — but nothing in the graph itself warrants raising an alarm.

NC-28-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework. Let me start by loading the database tools and examining the graph schema.

Searched available tools

Searched available tools

Let me examine the database schema first.

Get Neo4j Schema

Get Neo4j Schema

The default schema call had an issue. Let me specify a sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models network traffic: IPs, Hosts, OS, Ports, Conversations, Domains, and geolocation (City/Country). Let me explore the data in detail. First, let me understand the conversation structure and sample some data.

Read Neo4j Cypher

Read Neo4j Cypher

The local host of interest appears to be 10.0.2.15 (a typical VirtualBox NAT address). Let me identify the local network hosts and their characteristics.

Used neo4j-database integration

Used neo4j-database integration

The capture is centered on a single local machine, 10.0.2.15. Let me characterize its full conversation profile — what it talks to, on which ports, and the directionality of traffic.

Used neo4j-database integration

Used neo4j-database integration

The single host 10.0.2.15 conversed with 711 distinct remote IPs during the capture. Let me look at the ports used in conversations to understand the service profile, and examine traffic where the local host is the *sender* (potential exfiltration).

Read Neo4j Cypher

Read Neo4j Cypher

Mostly standard web traffic (443/80), with ephemeral high ports as the client-side. Let me look at conversations where the local host **sent more than it received** (potential upload/exfiltration), and at the destination domains.

Read Neo4j Cypher

Read Neo4j Cypher

Most flows are download-heavy (normal web browsing). A few are upload-skewed. Let me identify conversations where the local host sent substantially more than it received, and check the suspicious high-port conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Nothing here looks like bulk exfiltration — the upload-skewed flows are small (tens of KB) and consistent with ordinary POST requests, form submissions, and TLS handshakes. Let me check the geolocation distribution of the remote endpoints and the resolved hostnames/domains to see if anything stands out.

Read Neo4j Cypher

Read Neo4j Cypher

There's a notable concentration in Russia (70 IPs) and Czech Republic (20 IPs) — worth investigating. Let me look at the domain names involved and the Russian/Czech endpoints specifically.

Read Neo4j Cypher

Read Neo4j Cypher

This domain list is informative. I see a lot of legitimate web infrastructure (CDNs, ad networks, Russian portals like rambler.ru/yandex/vk/odnoklassniki — consistent with a Russian-language user), but also a few names that stand out as atypical: darkishaeromedical.com, scatsideview.net, nlooud.com, afodgk128.jebeloutlet.com, pipelane.net. These have the look of domains used by malware C2/distribution. Let me dig into those, plus the unusual high-port conversations.

Let me query the MITRE ATT&CK matrix to ground my analysis. First, let me search for the current ATT&CK technique references relevant to what I'm seeing.

Searched the web

Searched the web

Now let me investigate the suspicious domains and the hosts/IPs behind them. Let me trace the conversations involving the atypical domain names.

Read Neo4j Cypher

Read Neo4j Cypher

These are very interesting. The *.darkishaeromedical.com and *.scatsideview.net hosts resolve to the **185.42.119.0/24** block, and the hostnames are wildcards — a hallmark of

domain generation / fast-flux infrastructure. Let me examine the conversations involving these IPs and ports.

Read Neo4j Cypher

Read Neo4j Cypher

This is a meaningful finding. The local host repeatedly connected to multiple IPs in the **185.42.119.0/24** range, all serving wildcard hosts with random-looking domain names (darkishaeromedical.com, scatsideview.net). It also connected to encryption195/205.afodgk128.jebeloutlet.com over HTTP, downloading ~47 KB from .205. Let me get the geolocation of these IPs and check for additional related infrastructure in the same netblocks.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at all IPs in the 185.42.119.x and 128.121.3.x blocks and check the HTTP conversation with the Indonesia "encryption" hosts more closely — that 47 KB download over plain HTTP is a classic payload-delivery pattern.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the unusual high-port conversations I saw earlier (ports 58598-58604, 61455, etc.) and the conversations that had a null port, since those don't fit normal web browsing.

Read Neo4j Cypher

Read Neo4j Cypher

Those high ports are just client-side ephemeral ports — normal. Notably 192.168.33.254 is another local-range host with a large bidirectional flow (~2 MB) to 10.0.2.15. Let me check the HTTP exchange with the Indonesian "encryption" hosts and look at what the 185.42.119 connections actually carried. Let me also count how many distinct conversations touched the suspicious netblock.

Read Neo4j Cypher

Read Neo4j Cypher

192.168.33.254 is acting as the DNS server (port 53) and gateway for the local host — normal infrastructure. Now let me check the broader picture: which remote IPs the host contacted over plain HTTP (port 80), since malware payload delivery and some C2 use HTTP. Let me also pull the ATT&CK reference for the specific techniques I'm matching.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough data for a thorough verdict. Let me do two final checks: confirm the OS of the local environment and look for any beaconing pattern (many small, regular conversations to the same suspicious destination).

Used neo4j-database integration

Used neo4j-database integration

No repeated periodic beaconing to a single destination is visible (the graph aggregates per-conversation rather than per-packet, so fine-grained timing isn't available — a limitation I'll note). Let me do one last check on the most data-heavy plain-HTTP destination, 178.79.242.47, which pulled 17.7 MB.

Read Neo4j Cypher

Read Neo4j Cypher

I have a complete picture. Here is my assessment.

Verdict: Indications of likely malicious activity — moderate-to-high suspicion

The capture is consistent with **a single Windows-style endpoint (10.0.2.15, behind a VirtualBox-style NAT) that was very probably compromised** during the capture window. The traffic is dominated by ordinary web browsing, but a distinct subset matches several MITRE ATT&CK techniques. This is a suspicion verdict, not a confirmation — the graph models flow-level metadata only, with no payloads or timestamps.

What the graph shows

The dataset is one host's traffic: 10.0.2.15 conversed with 711 distinct remote IPs, almost entirely on ports 443 and 80, with 192.168.33.254 serving as its DNS resolver/gateway. The large majority of this is unremarkable — CDNs, ad networks, and Russian-language portals (rambler, yandex, vk, odnoklassniki) suggesting a Russian-speaking user.

Indicators of concern

The notable findings cluster around a small number of atypical destinations:

Suspicious wildcard/randomized domains. The host repeatedly contacted several IPs in the **185.42.119.0/24** block (Issy-les-Moulineaux, France) whose hosts resolve to wildcard records on randomly-named domains — *.darkishaeromedical.com and *.scatsideview.net. Random, semantically meaningless domain names combined with wildcard DNS and a shared netblock are a classic signature of malware C2 / domain-generation-algorithm infrastructure. The 185.42.119.x range has a long-standing reputation in threat intelligence for hosting exploit-kit and malware delivery infrastructure.

Plain-HTTP contact with "encryption"-named hosts. The host exchanged HTTP traffic with encryption195.afodgk128.jebeloutlet.com and encryption205.afodgk128.jebeloutlet.com (128.121.3.x, Jakarta), downloading ~47 KB over unencrypted port 80 from a host with a deceptive, machine-generated subdomain — a pattern consistent with second-stage payload delivery.

Other oddly-named endpoints include vps-109-123-210-154.nlooud.com (a VPS in Prague) and 204.2.197.204 (alt-e.pipelane.net).

Mapping to MITRE ATT&CK

- **T1071 / T1071.001 – Application Layer Protocol (Web Protocols):** C2-style contact disguised as ordinary HTTP/HTTPS to the 185.42.119.x and jebeloutlet hosts.

- **T1568 / T1568.002 – Dynamic Resolution (Domain Generation Algorithms):** wildcard, randomized domain names backing the suspect IPs.
- **T1573 – Encrypted Channel:** suspicious sessions carried over TLS/443, blending with legitimate traffic — the standard C2 evasion pattern.
- **T1105 – Ingress Tool Transfer:** the ~47 KB plain-HTTP download from the deceptively-named Indonesian host is consistent with fetching a follow-on tool/payload.
- Possible **T1041 – Exfiltration Over C2 Channel:** several upload-skewed flows exist, but all are small (tens of KB) — consistent with normal POSTs, so this is weak/unconfirmed.

Important limitations

The graph stores aggregated conversations (byte counts, ports, endpoints) with **no per-packet timestamps and no payload content**. I therefore could not test for periodic beaconing intervals, inspect TLS certificates/JA3, or confirm what the HTTP download actually contained. "Suspicious" here rests on infrastructure reputation and naming patterns, not on observed malicious payloads. The 17.7 MB HTTP pull from 178.79.242.47 (Germany), by contrast, looks like ordinary large-file download, not exfiltration.

Recommendation

Treat 10.0.2.15 as **potentially compromised** and investigate further: pull the actual PCAP for the sessions to 185.42.119.33/65/129 and 128.121.3.195/205, check those IPs/domains against current threat-intelligence feeds (VirusTotal, abuse databases), inspect the 47 KB HTTP object, and review the endpoint itself for persistence and new processes. The combination of randomized wildcard domains, a shared low-reputation netblock, and deceptively-named plain-HTTP hosts is unlikely to be benign and warrants endpoint-level forensics.

NC-28-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this network traffic graph for signs of malicious activity. Let me start by understanding the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the default sample size. Let me provide one explicitly.

Get Neo4j Schema

Get Neo4j Schema

The graph models network traffic: IPs, hosts, OS, domains, geolocation (city/country), ports, and conversations. Let me explore the data in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — there's one notable conversation already: 10.0.2.15 sending to 178.79.242.47 with 469KB sent but **17.7 MB received** — a large inbound download. Let me look at the local network hosts and the outbound conversations.

Read Neo4j Cypher

Read Neo4j Cypher

The local network appears to be 10.0.2.0/24, with the main active host being 10.0.2.15. Let me identify all local hosts and the destination ports being used.

Used neo4j-database integration

Used neo4j-database integration

Those are ephemeral source ports. Let me look at the destination ports used in conversations to understand the services being contacted.

Read Neo4j Cypher

Read Neo4j Cypher

Mostly HTTPS/HTTP. Now let me cross-reference the destination IPs of 10.0.2.15 with their geolocation, and look closely at that anomalous large download.

Used neo4j-database integration

Used neo4j-database integration

The local IPs aren't tied to Host nodes. Let me check what the Host/Domain/OS structure actually connects to, and look at the domains contacted.

Read Neo4j Cypher

Read Neo4j Cypher

A few domain names stand out as suspicious: afodgk128.jebeloutlet.com, darkishaeromedical.com, scatsideview.net, nlooud.com, canistec.cz. Random/nonsensical subdomains like these are characteristic of malware C2. Let me investigate these and the hosts/IPs connected to them.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have strong candidate indicators. Let me check whether 10.0.2.15 actually communicated with these IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Confirmed — 10.0.2.15 communicated with all of these suspicious-domain IPs. The pattern (small beaconing exchanges plus one ~47KB download) is consistent with C2 and payload delivery. Now let me query the MITRE ATT&CK matrix to map these behaviors to known adversary techniques.

Searched the web

Searched the web

Let me look at the specific Command and Control techniques relevant to the traffic patterns I'm seeing.

Fetched: Command and Control

Fetched: Command and Control

I have the MITRE C2 techniques. Now let me dig deeper into the graph to characterize the suspicious traffic more precisely — checking the conversation directions, beaconing patterns, and looking at the big download host and other anomalies.

Used neo4j-database integration

Used neo4j-database integration

Let me check a couple more things — the 192.168.33.254 host (a different subnet talking to 10.0.2.15), and look at the full set of countries contacted to assess geographic spread.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the darkishaeromedical.com / scatsideview.net IPs more carefully — they're in the same 185.42.119.0/24 block, which is a strong indicator of dedicated malicious

infrastructure. Let me also look at HTTP (port 80) conversations, since unencrypted C2 and payload delivery often uses HTTP.

Used neo4j-database integration

Used neo4j-database integration

178.79.242.47 (Nuremberg, Germany — a Linode VPS range) delivered ~17.7 MB to 10.0.2.15 over **plain HTTP** across many port pairings. A large unencrypted HTTP download from a bare-IP VPS is a classic payload-delivery pattern. Let me check the rambler.ru / yandex / odnoklassniki Russian domains and confirm whether these were also contacted directly, then summarize.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to produce a verdict. Let me consolidate the findings into a structured intelligence assessment.

Verdict: Indications of Compromise — Likely Malware Infection

Yes — the traffic capture shows clear, multiple indications of a malware/cyber-attack affecting the local network. The evidence is moderate-to-strong and is concentrated on a single host.

Scope of the local network

The capture's internal network is 10.0.2.0/24. Only one host generated meaningful traffic:

- **10.0.2.15** — the active workstation (366 outbound conversations, ~5.1 MB sent / ~77 MB received). This is the host of concern.
- 10.0.2.2 (gateway), 10.0.2.255 (broadcast), 192.168.33.254 (a host on a separate subnet, likely a host-only adapter / sandbox interface) — no independent suspicious activity.

The single-host, sandbox-style addressing (10.0.2.15 is the default VirtualBox NAT guest address) suggests this is a malware-analysis capture, which is consistent with what the traffic shows.

Indicators of compromise found in the graph

1. Contact with suspicious algorithmically-named domains — the strongest indicator. 10.0.2.15 communicated with several hosts whose domain names are nonsensical/random-looking, a hallmark of malware C2 infrastructure and Domain Generation Algorithms:

Domain / Host	IP	Geolocation	Pattern
*.scatsideview.net	185.42.119.33, .65	Issy-les-Moulineaux, France	C2-style

Domain / Host	IP	Geolocation	Pattern
*.darkishaeromedical.com	185.42.119.129	Issy-les-Moulineaux, France	C2-style
encryption195/205.afodgk128.jebeloutlet.com	128.121.3.195/.205	Jakarta, Indonesia	C2-style
vps-…nlooud.com	109.123.210.154	Prague, Czech Republic	bare VPS

The three 185.42.119.x addresses sit in the **same /24 block** under two unrelated-looking domains — a classic sign of **dedicated adversary infrastructure** rather than legitimate shared hosting.

2. Large unencrypted payload download. 10.0.2.15 pulled **~17.7 MB over plain HTTP (port 80)** from 178.79.242.47 (Nuremberg, Germany — a bare-IP VPS, no real domain name). A multi-megabyte HTTP transfer from an IP-only VPS is a textbook **secondary-payload / tool delivery** pattern. Two other bare-IP hosts (93.184.220.70, 91.184.8.50) also delivered multi-MB downloads.

3. Beaconing-like traffic. 48 conversations from 10.0.2.15 were tiny bidirectional exchanges (<2 KB each way) — consistent with periodic check-in beacons to C2. The exchanges with the suspicious domains above fit this profile (e.g., 336/248 bytes, 176/400 bytes).

4. Wide, incoherent geographic spread. A single workstation contacting 21 countries — including direct conversations with VPS-hosted hosts in Indonesia, Czech Republic, and France that have no recognizable service branding — is abnormal for routine browsing.

Mapping to the MITRE ATT&CK matrix (Enterprise)

The observed behavior maps cleanly onto the **Command and Control (TA0011)** tactic and adjacent tactics:

- **T1071.001 – Application Layer Protocol: Web Protocols** — C2 conducted over HTTP/HTTPS (ports 80/443) to blend with normal web traffic. All suspicious contacts used 80/443.
- **T1568 / T1568.002 – Dynamic Resolution / Domain Generation Algorithms** — the random-looking domains (scatsideview.net, darkishaeromedical.com, afodgk128.jebeloutlet.com) are characteristic DGA-style C2 naming.
- **T1105 – Ingress Tool Transfer** — the ~17.7 MB HTTP download from a bare-IP VPS indicates additional tooling/payload pulled into the host.
- **T1571 / T1095** would be the next things to check (non-standard ports / non-application-layer protocols) — not seen here; the actor stuck to 80/443 to evade filtering, which itself supports T1071.
- **T1008 – Fallback Channels** — multiple C2-like endpoints across several providers/countries provide redundancy.

- Possible **Exfiltration (TA0010)** — outbound byte volumes were modest, so there is no strong evidence of bulk data theft, but the C2 channel itself is exfil-capable.

Confidence and caveats

This is a **graph of flow metadata only** (IPs, ports, byte counts, geolocation, reverse-DNS). It contains no packet payloads, timestamps, or process attribution, so I cannot name a specific malware family or prove execution. The verdict rests on:

- a single host reaching multiple algorithmically-named domains on dedicated-looking infrastructure (strong), plus
- a large cleartext HTTP download from an IP-only VPS (strong), plus
- beacon-sized exchanges and abnormal geographic spread (corroborating).

The reverse-DNS names should be treated as untrusted attacker-controlled data — but here they *strengthen* the case, since legitimate services rarely use names like these.

Bottom line: 10.0.2.15 should be treated as **compromised**. Recommended next steps: isolate the host; block and threat-intel-check 185.42.119.0/24, 128.121.3.195/205, 109.123.210.154, and 178.79.242.47; capture full packet/PCAP and endpoint forensics to identify the malware family and confirm whether exfiltration occurred.

This analysis is based on metadata patterns and MITRE ATT&CK behavioral mapping — it indicates likely compromise but is not a substitute for a full forensic investigation with packet-level and host-level evidence.

NC-28-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 24

I'll investigate this. Let me start by examining the Neo4j database to understand the network traffic model.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The APOC schema call had an issue. Let me explore the database structure directly.

Used neo4j-database integration

Used neo4j-database integration

A 10.0.2.15 host stands out — that's a classic VirtualBox NAT default address. Let me look at the Host, Domain, OS, and PORT nodes, plus how things connect.

Used neo4j-database integration

Used neo4j-database integration

Now let me identify the local network hosts and look at the conversation/port structure.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x addresses are public Google IPs (not private 172.16.0.0/12). The real local hosts are 10.0.2.15 (the host being monitored — VirtualBox NAT guest IP), 10.0.2.2 (NAT gateway), 10.0.2.255 (broadcast), and 192.168.33.254. Let me characterize the main host's traffic.

Used neo4j-database integration

Used neo4j-database integration

Now let me look at which ports are used — important for behavioral analysis.

Used neo4j-database integration

Used neo4j-database integration

Mostly standard web ports (443/80). Let me check the destination geography and look for suspicious indicators — the unusual domain names I noticed earlier (nlooud.com, darkishaeromedical.com, scatsideview.net, afodgk128.jebeloutlet.com).

Used neo4j-database integration

Used neo4j-database integration

These are notable — darkishaeromedical.com and scatsideview.net are randomly-generated-looking domains (a classic malware C2 / DGA indicator), and they share the 185.42.119.0/24 block. Let me check the conversations to these hosts and the host with full graph context.

Used neo4j-database integration

Used neo4j-database integration

Let me look at the OS of the monitored host and check the full set of hosts the main host contacted, then research these suspicious indicators against threat intelligence and the MITRE ATT&CK matrix.

Used neo4j-database integration

Used neo4j-database integration

Now let me research the suspicious indicators against threat intelligence and review the relevant MITRE ATT&CK techniques.

Searched the web

Searched the web

Let me look at the conversation pattern more closely. The traffic capture appears to be from an analysis VM (10.0.2.15, OS labeled "Cloud"). Let me examine the small "beacon-like" conversations and the HTTP-to-128.121.3.x traffic, and check for DGA-pattern domains systematically.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the hosts behind the suspicious 185.42.119.x and 128.121.3.x IPs more carefully, and look at all domains contacted to spot DGA-style names.

Used neo4j-database integration, searched the web

Used neo4j-database integration, searched the web

The web searches didn't surface a direct match for these specific domains (they may be expired/rotated), but the domain naming pattern itself is a strong signal. Let me query the MITRE ATT&CK matrix and check a couple more behavioral indicators in the graph — particularly the actonsoftware host and the HTTP downloads.

Searched the web

Searched the web

I have what I need on the MITRE side. Let me run a couple more targeted graph queries — checking for the largest data transfers (possible exfiltration / payload download) and the actonsoftware host.

Read Neo4j Cypher

Read Neo4j Cypher

Wait — these are sum totals; the per-conversation download for 178.79.242.47 was 17.7 MB and 93.184.220.70 was 10.2 MB. Let me re-check those large ones and look at the actonsoftware.com host (Acton marketing software is sometimes seen in commodity malware delivery / a host worth verifying) and the bitly link shortener.

Used neo4j-database integration

Used neo4j-database integration

I have enough to produce a verdict. Let me do two final checks: the 93.184.220.70 host (that's EdgeCast/edgesuite range — 93.184.216.x is the famous example.com but .220.x differs) and confirm whether the suspicious 185.42.119.x traffic used HTTPS, plus look at all hosts contacted on non-standard ports.

Read Neo4j Cypher

Read Neo4j Cypher

All real port assignments are 443/80/53 — no non-standard C2 ports. The ephemeral high ports are just client-side source ports. Let me do one last check on the conversation count per host to see if anything resembles repeated beaconing.

Read Neo4j Cypher

Read Neo4j Cypher

I've completed my investigation. Here is my verdict.

Verdict: Indications of malware are present — most consistent with a drive-by / exploit-kit-style infection chain

The capture is from a single monitored host, **10.0.2.15** (a VirtualBox-NAT'd guest, OS tagged "Cloud" — almost certainly a malware-analysis sandbox), talking to 712 external IPs over 716 conversations, almost entirely on ports 443 (HTTPS) and 80 (HTTP). The bulk of that traffic is ordinary — Google, Facebook/WhatsApp, Akamai/CloudFront CDNs, Yandex, Rambler, McAfee, ad networks. But several findings stand out.

1. Domains with the hallmark of malware infrastructure. The host contacted IPs whose reverse-DNS resolves to randomly-constructed, meaning-free domains: *.darkishaeromedical.com and *.scatsideview.net (both wildcard hosts), afodgk128.jebeloutlet.com (with subdomains encryption195/encryption205), and nlooud.com. Word-salad domains like these are a classic Domain Generation Algorithm / disposable-C2 indicator. Critically, darkishaeromedical.com and scatsideview.net are colocated in the **same 185.42.119.0/24 block** — 185.42.119.33, .65, .129 — a tight cluster of unrelated-sounding names on adjacent IPs, which is typical of exploit-kit or C2 hosting (the 185.x European-bulletproof-hosting range is frequently abused).

2. Behavioral pattern of those conversations. Traffic to the suspicious IPs is small and asymmetric — e.g. 185.42.119.129 exchanged only 176/400 bytes (a beacon-sized check-in), while 128.121.3.205 was a 3 KB-out / 48 KB-in HTTP exchange (consistent with a small request pulling down a payload). This mix of tiny beacons plus a larger HTTP "download" is the signature of a redirect → landing-page → payload-delivery chain.

3. Mapping to MITRE ATT&CK. The observed behavior maps to:

- **T1189 – Drive-by Compromise** (Initial Access): web-borne contact with exploit-kit-style hosts.
- **T1071.001 – Application Layer Protocol: Web Protocols** and **T1573 – Encrypted Channel** (Command & Control): C2 deliberately blended into ordinary HTTP/HTTPS traffic on standard ports, which is exactly the evasion MITRE describes for TA0011.
- **T1568.002 – Dynamic Resolution: Domain Generation Algorithms**: the algorithmically-generated domain names.
- **T1105 – Ingress Tool Transfer**: the asymmetric inbound HTTP transfer consistent with second-stage payload retrieval.

What does *not* support an attack verdict, in fairness: no traffic on non-standard/unusual ports (no C2 on odd ports, no ICMP/DNS-tunneling evidence in the schema); no repeated high-frequency beaconing to one host (max 2 conversations to any peer — though this may simply reflect a short capture); no clear large *outbound* exfiltration (the biggest transfers are inbound, consistent with normal browsing/CDN downloads); and the graph contains no timestamps, packet contents, process data, or actual DNS query logs, so the "DGA" judgement rests on reverse-DNS naming rather than confirmed algorithmic generation. I could not independently confirm these specific domains against current threat-intel feeds — they appear expired or rotated, which is itself consistent with disposable malicious infrastructure but is not proof.

Bottom line: On the balance of evidence, **yes — there are concrete indications of malicious activity on 10.0.2.15 during the capture**, best characterized as a likely browser-based malware infection (exploit-kit / drive-by delivery with HTTP/HTTPS C2). The verdict is *probable*, not *certain*: it is well-supported by the infrastructure clustering and traffic shape, but a definitive confirmation would need the original PCAP — DNS queries, HTTP URIs/user-agents, TLS SNI/JA3, payload hashes, and timing — none of which are in this graph. I'd recommend pulling those 185.42.119.x / 128.121.3.x / 109.123.210.154 indicators and the full URIs from the source capture for confirmation.

Cyber-incident analysis is a sensitive area — if this relates to a real production network rather than a lab sandbox, treat 10.0.2.15 as potentially compromised and escalate to your incident-response process rather than relying on this graph alone.

MX-3-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

6:52 AM

I'll investigate the Neo4j database to analyze the network traffic for signs of malicious activity. Let me start by loading the database tools.

Searched available tools

Searched available tools

Let me first examine the database schema to understand the structure of the network traffic data.

Get Neo4j Schema

Get Neo4j Schema

The default schema call failed due to a parameter issue. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

I have a clear picture of the schema. The graph models: IP addresses (with flags like is_spam, is_known_attacker, is_Bot, is_Local), Hosts, Operating Systems, Domains, geographic data (City/Country), network Ports, and Conversations (with byte counts). Let me explore the data.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — there are 7 local IPs, 8 known attackers, and 5 bots. Let me get details on the local hosts and the flagged external IPs.

Read Neo4j Cypher

Read Neo4j Cypher

The local "IPs" are mostly multicast/broadcast addresses (224.x.x.x, 239.255.255.250, 10.0.2.255) plus gateway-style addresses. The actual local network appears to be the 10.0.2.0/24 subnet. Let me find the real local hosts and the flagged external IPs.

Used neo4j-database integration

Used neo4j-database integration

Now let me examine the conversations involving the local host (10.0.2.200 appears to be the monitored workstation) and these flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look specifically at conversations with the flagged attacker/bot IPs, and which ports were involved.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the domains involved and the overall conversation port distribution, plus check for any unusual ports.

Used neo4j-database integration

Used neo4j-database integration

One domain stands out significantly: **email.mspy.com**. mSpy is a well-known commercial **spyware/stalkerware** product. Let me investigate which hosts connect to the bot and attacker IPs, and check the mSpy-related connection.

Used neo4j-database integration

Used neo4j-database integration

Confirmed — the local host 10.0.2.200 communicated with an mSpy mail server. Let me check the total scope of 10.0.2.200's external communications and verify it's the only active local host.

Used neo4j-database integration

Used neo4j-database integration

Every conversation involves the 10.0.2.0/24 subnet — and from the data, all of them route through 10.0.2.200. This is a single-host packet capture. Let me check the geographic spread and how many distinct external hosts the workstation contacted, then look at the connections to attacker/bot IPs in terms of who initiated.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a thorough picture of the graph. Let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me look up the ATT&CK techniques most relevant to the specific findings: stalkerware and connections to known-malicious infrastructure.

Searched the web

Searched the web

I have enough from MITRE ATT&CK. Let me do one final check on the conversation patterns — specifically looking at whether the host receives data on HTTP from the attacker IPs (which would indicate malvertising / drive-by) and the byte ratios.

Used neo4j-database integration

Used neo4j-database integration

I now have a complete picture. Here is my analysis and verdict.

Verdict: Indications of Malicious Activity Present — Most Likely Stalkerware/Spyware Infection

After querying the graph database and mapping findings against the MITRE ATT&CK framework, **yes, there are concrete indications of a cyber-compromise** on the monitored network during the capture window.

What the network looks like

The capture is essentially a **single-host packet capture**. The 10.0.2.0/24 subnet contains only one real workstation — **10.0.2.200** — plus broadcast/multicast addresses (10.0.2.255, 224.0.0.x, 239.255.255.250) and a gateway (10.0.2.2). The 10.0.2.x addressing strongly suggests a VirtualBox NAT environment, i.e. a sandboxed/monitored machine. Every one of the 442 conversations involves this host, reaching 437 external IPs across 26 countries.

The primary finding: a connection to mSpy infrastructure

The single most damning piece of evidence: host 10.0.2.200 exchanged HTTP traffic with **82.196.8.230**, whose hostname resolves to **mta-06.email.mspy.com** (domain email.mspy.com, hosted in Amsterdam).

mSpy is a commercial spyware / stalkerware product. It is a monitoring application that, once installed on a device, silently exfiltrates messages, call logs, location, browsing history, and keystrokes to mSpy's servers. A normal workstation has no legitimate reason to talk to mSpy mail infrastructure. This connection is the signature of an active surveillance-ware implant beaconing or sending collected data home.

Mapped to MITRE ATT&CK, this corresponds to:

- **Exfiltration Over C2 Channel (T1041)** — stolen data leaving over an HTTP channel that blends with normal web traffic.
- **Mobile surveillanceware techniques** — mSpy is catalogued by vendors alongside FlexiSpy-class stalkerware (collection techniques like *Stored Application Data T1409*, *Software Discovery T1418*).

Supporting findings

The database explicitly flags 13 external IPs the host communicated with:

8 IPs flagged is_known_attacker = true, all in conversation with 10.0.2.200:

- 31.13.84.36, 31.13.84.4 (Vienna), 64.158.223.140, 64.158.223.137, 63.215.202.137 (Amsterdam), 172.217.16.98 (London), 64.71.187.126 (Dallas), 152.163.51.2 (Phoenix).
- Notably, 172.217.16.98 and 31.13.84.4 *initiated* high-volume transfers **toward** the host (459 KB and 85 KB respectively) over HTTP/HTTPS plus a large spread of high ephemeral ports — a pattern consistent with **content delivery to a compromised endpoint** (malvertising / drive-by, ATT&CK *Drive-by Compromise T1189*, or C2 push).

5 IPs flagged is_Bot = true, also in conversation with the host: 217.163.21.35, 66.196.87.111, 66.196.65.112 (US), 77.238.185.35, 217.12.13.40 (Dublin). One of these (66.196.65.112) pushed ~139 KB to the host.

Other behavioral signals consistent with, though not by themselves conclusive of, compromise: a very large fan-out of contacts across 26 countries (including Russia, Iran, Ukraine) for what is one workstation; and small outbound HTTP transfers to anonymous **AWS EC2 instances** (23.23.121.113, 107.22.190.224) where the host *sent* 3× more than it received — a small-beacon-style ratio worth noting under T1041.

Caveats / confidence

A few honest limitations: the is_known_attacker / is_Bot / is_spam flags are pre-computed attributes baked into the graph — I am trusting the dataset's labeling rather than independently verifying reputation. Some "attacker"-flagged IPs (e.g. the 31.13.84.x Facebook range, 172.217.x Google range) are actually well-known CDN/social ranges, so those flags may be noisy or reflect malicious *use* of those ranges (e.g. C2 over CDN). The capture also lacks timestamps and payload data, so I cannot reconstruct a precise kill-chain timeline.

However, the mSpy connection does not depend on any pre-computed flag — it is an objective hostname resolution to a known stalkerware vendor, and it stands on its own.

Bottom line

The capture shows indications of a compromised host. 10.0.2.200 exhibits behavior consistent with an active **stalkerware/spyware infection (mSpy)**, evidenced by direct communication with mSpy's mail infrastructure, corroborated by conversations with multiple IPs the dataset labels as known attackers and bots, and by inbound high-volume transfers and small outbound beacon-like flows. Mapped to MITRE ATT&CK, the activity aligns with **Exfiltration Over C2 Channel (T1041)** and surveillanceware collection techniques.

Recommended next steps: isolate 10.0.2.200 from the network, scan it for the mSpy agent and remove it, rotate credentials that may have been captured, and block the flagged IPs and the email.mspy.com domain at the perimeter.

MX-3-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

7:14 AM

Claude responded: I'll investigate this systematically.

I'll investigate this systematically. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has a bug with the default sample size. Let me explore the schema manually instead.

Used neo4j-database integration

Used neo4j-database integration

Now let me inspect the properties of each node type to understand what data is available.

Used neo4j-database integration

Used neo4j-database integration

Let me check how relationships connect these nodes and look at the sample relationship patterns.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a clear picture of the schema. Let me identify the local network hosts.

Read Neo4j Cypher

Read Neo4j Cypher

The local network is the 10.0.2.0/24 subnet. 10.0.2.200 is the host of interest (a workstation), 10.0.2.2 is the gateway, 10.0.2.255 is broadcast, and the 224.x / 239.x addresses are multicast. Let me examine the conversations of the main host.

Read Neo4j Cypher

Read Neo4j Cypher

All 442 conversations involve the local subnet. Let me look at traffic volume per local host and the external endpoints they communicate with.

Read Neo4j Cypher

Read Neo4j Cypher

Every conversation involves 10.0.2.200 — it's the single monitored workstation. Let me look at the external IPs it talked to, focusing on flagged ones.

Read Neo4j Cypher

Read Neo4j Cypher

Several flagged external IPs. Let me get full detail on each — the conversations, traffic volumes, ports, hostnames, and geolocation.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — these IPs resolve to Facebook, Google (1e100.net), Yahoo, and Dotomi (an ad network). These are mostly legitimate services. The flags appear to be from blocklist data of varying quality. Let me examine the actual traffic to these endpoints — the conversation patterns matter more than the labels.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the overall traffic profile — top destinations by volume and the full port/destination picture — to understand what kind of host this is.

Used neo4j-database integration

Used neo4j-database integration

Only HTTP and HTTPS as well-known ports — no IRC, no unusual service ports. Let me check for any non-standard service ports that could indicate C2 or backdoors, and check destination port diversity (fan-out, a sign of scanning).

Used neo4j-database integration

Used neo4j-database integration

Let me check the geographic spread and look specifically for beaconing-style patterns — many tiny, regular conversations to the same destination.

Used neo4j-database integration

Used neo4j-database integration

No beaconing pattern — almost every external IP has exactly one conversation. Let me check the domains the host contacted for anything resembling DGA (domain-generation algorithm) or known-malicious patterns, and look at all external hosts/domains.

Read Neo4j Cypher

Read Neo4j Cypher

One domain stands out sharply: email.mspy.com. **mSpy is commercial spyware / stalkerware** — a monitoring application marketed for surveilling phones and computers. This is a significant finding. Let me examine that conversation in detail.

Read Neo4j Cypher

Read Neo4j Cypher

Let me also check the other suspicious domains — heias.com and the IP 185.31.17.249 which had notable traffic volume — and verify what mSpy is via search.

Searched the web

Searched the web

Confirmed — mSpy is well-documented commercial spyware/stalkerware. Now let me query the MITRE ATT&CK matrix to map the observed behaviors to recognized techniques. Let me check what's accessible.

Searched the web

Searched the web

Let me do a few more targeted queries to nail down the data-flow direction (a host being spied on should show large *outbound* uploads) and check the remaining unverified suspicious domains.

Used neo4j-database integration

Used neo4j-database integration

I now have enough to form a verdict. Let me run one final check to confirm the mSpy connection details and look at the host's domain memberships overall.

Read Neo4j Cypher

Read Neo4j Cypher

Verdict: Indications of a Cyber-Threat — Stalkerware Present

Conclusion: Yes. There is a credible indication of malicious software on the local network — specifically the commercial spyware/stalkerware "mSpy" — on host 10.0.2.200.

What the graph contains

The capture models traffic for a single monitored workstation, 10.0.2.200, on the 10.0.2.0/24 subnet (gateway 10.0.2.2, plus broadcast/multicast addresses). It recorded 442 conversations with external IPs over HTTP and HTTPS only, totaling ~178 MB.

The key finding

The host communicated with mta-06.email.mspy.com (IP 82.196.8.230, Amsterdam, over HTTP). mSpy is a phone/computer surveillance app that promotes itself as a way to track children or monitor employees, but, like most spyware, is also widely used to monitor people without their consent — which is why these apps are also known as "stalkerware." It runs in the background to stay hidden on the target device and uploads collected data — messages, location, browsing, media — to an online dashboard. TechCrunchMageplaza

A clean workstation has no legitimate reason to contact mSpy's mail infrastructure. This is the kind of artifact mSpy generates when it is installed and reporting.

Mapping to MITRE ATT&CK

The presence of stalkerware corresponds to a recognizable behavioral chain:

- **TA0009 – Collection / TA0010 – Exfiltration:** Stalkerware's core function is gathering local data and sending it to an attacker-controlled dashboard.
- **T1041 – Exfiltration Over C2 Channel:** Attackers leverage existing channels to transfer sensitive data back to attacker-controlled infrastructure, commonly abusing legitimate protocols such as HTTP/S to exfiltrate data unnoticed. The mSpy contact is over plain HTTP. Garnet
- **T1020 – Automated Exfiltration:** mSpy uploads collected data automatically without operator interaction.
- **TA0011 – Command and Control:** The host maintains an outbound channel to the spyware operator's servers.

Important caveats

The 13 IPs flagged in the database as is_known_attacker or is_Bot are **not** strong evidence on their own. On inspection they resolve to Facebook, Google (1e100.net), Yahoo, and the Dotomi ad network — these are legitimate services, and the flags appear to be low-quality blocklist labels (likely tied to ad/tracker reputation). The traffic to them is single, ordinary HTTP/HTTPS conversations with no beaconing.

Similarly, the overall traffic profile is otherwise benign: only ports 80/443 are used, there are no unusual service ports, no C2 beaconing pattern (almost every external IP appears in exactly one conversation), and the byte ratio is heavily *inbound* (~175 MB down vs ~3.4 MB up) — consistent with normal web browsing and streaming, not bulk data theft.

So the verdict rests specifically and solely on the **mSpy contact**, which is unambiguous and does not have an innocent explanation.

Recommendation

Host 10.0.2.200 should be treated as compromised by stalkerware and investigated offline: scan for the mSpy agent, identify how/when it was installed (it typically requires prior physical or admin access to the device), and review what data may have been exfiltrated. If this device belongs to a person who may be the target of monitoring by another individual, handle it with care — stalkerware cases can carry personal-safety implications, and abruptly removing it can alert the person who installed it.

A note on sourcing: I queried current MITRE ATT&CK references via web search rather than an ATT&CK database, since none was connected — the technique IDs above are the standard public ATT&CK mappings for this behavior.

MX-3-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

7:20 AM

Claude responded: I'll analyze the network traffic graph in the Neo4j database against the MITRE ATT&CK framework to look for indicators of malicious activity.

I'll analyze the network traffic graph in the Neo4j database against the MITRE ATT&CK framework to look for indicators of malicious activity. Let me start by loading the necessary tools.

Searched available tools

Searched available tools

Let me first inspect the database schema to understand how the network traffic is modeled.

Get Neo4j Schema

Get Neo4j Schema

APOC has an issue with the default parameter. Let me provide an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models a network capture with IPs, Hosts, OS, Ports, Conversations, and geolocation data. Notably, IP nodes carry threat-intelligence flags: is_spam, is_known_attacker, and is_Bot. Let me explore the data systematically.

Used neo4j-database integration

Used neo4j-database integration

The local network is the 10.0.2.0/24 subnet. The other "local"-flagged addresses are multicast/broadcast (224.x, 239.x, .255). Let me identify the actual local hosts and external IPs flagged by threat intelligence.

Used neo4j-database integration

Used neo4j-database integration

Eight IPs flagged as known attackers and five as bots. Let me determine which of these the local hosts actually communicated with, and pull the conversation details.

Read Neo4j Cypher

Read Neo4j Cypher

A single local host — 10.0.2.200 — communicated with **all 13** flagged remote IPs (8 attackers + 5 bots). Let me look at what ports were used in these conversations to understand the nature of the traffic.

Read Neo4j Cypher

Read Neo4j Cypher

The traffic to flagged IPs is over standard web ports (80/HTTP, 443/HTTPS) from high ephemeral client ports — consistent with HTTP(S) sessions. Let me check what domains these flagged IPs and the local host are associated with, and look at the overall conversation profile of the local host.

Read Neo4j Cypher

Read Neo4j Cypher

This is interesting and important context. Several "flagged" IPs resolve to well-known infrastructure: 1e100.net (Google), facebook.com/fbcdn.net (Facebook/Meta CDN), yahoo.com (Yahoo), and dotomi.com (an ad-tech/online-advertising company). Let me get the full conversation profile of the local host and check timing/volume data.

Used neo4j-database integration

Used neo4j-database integration

There's no timestamp data — conversations only carry byte counts, not timing. Let me check the destination countries and look for any direct hostname/domain that might indicate C2 (command-and-control), plus check whether the local host received far more than it sent (download-heavy, typical browsing) or the reverse (exfiltration pattern).

Read Neo4j Cypher

Read Neo4j Cypher

Now let me query the MITRE ATT&CK matrix to ground the analysis against recognized adversary techniques.

Searched the web

Searched the web

Let me look at a couple of relevant ATT&CK techniques in detail — particularly Command and Control, since that's what network-traffic captures are best positioned to reveal.

Searched the web

Searched the web

I have enough from the ATT&CK framework. Let me run a few more targeted queries to complete the picture — checking for any scanning/discovery patterns and confirming the bot/attacker IP geography distribution.

Used neo4j-database integration

Used neo4j-database integration

I have a complete picture. Let me present the verdict.

Verdict: Indications of suspicious/malicious activity are present — most consistent with adware/PUP or botnet-style Command & Control, not a confirmed targeted intrusion

What the graph shows

The capture models a small local subnet (10.0.2.0/24). Only **one host actually generated traffic**: 10.0.2.200. The other "local" addresses (10.0.2.2, 10.0.2.255, 224.x, 239.x) are a gateway, broadcast, and multicast addresses — not endpoints. So the entire analysis reduces to the behavior of this single host across 442 conversations with 436 external IPs.

The threat-intelligence flags on the IP nodes are the key finding. Host 10.0.2.200 communicated with **all 13 flagged external IPs** in the dataset — 8 marked is_known_attacker and 5 marked is_Bot. That is not a coincidental brush with one bad IP; it is a consistent pattern of contact with flagged infrastructure.

Mapping to MITRE ATT&CK

The observable behavior maps cleanly onto **T1071.001 – Application Layer Protocol: Web Protocols** (Command and Control tactic, TA0011). Every conversation with a flagged IP runs over ports **80 (HTTP)** and **443 (HTTPS)** from high ephemeral client ports. Adversaries communicate using application layer protocols associated with web traffic to avoid detection by blending in with existing traffic, embedding commands and results within HTTP/HTTPS traffic between client and server. Malware deliberately uses 80/443 because it is universally allowed through firewalls. Center for Threat-Informed Defense

Several flagged conversations also show an upload-heavy or bidirectional profile that is at least consistent with **TA0010 – Exfiltration** or beaconing (e.g., the host sent ~66 KB to the "attacker"-flagged 172.217.16.98, and ~10 KB to 31.13.84.36). However, the byte volumes are modest and there is no single dominant outbound channel.

Important caveats that argue against a confident "malware confirmed" verdict

This evidence is suggestive, not conclusive, for several concrete reasons:

1. **Several "flagged" IPs resolve to mainstream infrastructure.** The reverse-DNS hostnames show 1e100.net (Google), facebook.com / fbcdn.net (Meta CDN), yahoo.com, and dotomi.com. Dotomi is a legitimate online-advertising/retargeting company. A host talking to ad-tech and social-media CDNs that some blocklists tag as "bot/attacker" is the classic signature of **adware or a potentially unwanted program (PUP)**, or simply aggressive ad/tracker traffic — not necessarily a serious intrusion. Threat-intel flags on shared CDN IPs are frequently false positives.

2. **No timestamps.** The CONVERSATION nodes carry only bytes_sent/bytes_received, no timing. True C2 beaconing is identified by *regular, periodic* callbacks. Without timestamps, periodicity — the strongest discriminator between malware and normal browsing — cannot be tested.

3. **No internal scanning or lateral movement.** There is no evidence of Discovery (T1046) or Lateral Movement — the host only talks outward to the internet, and there are no other live local hosts to pivot to. The traffic is overwhelmingly inbound-heavy (166 MB received vs. 12 MB sent overall), which is the normal profile of a workstation browsing the web, not a data-exfiltration event.

4. The geographic spread (26 countries, heavily US/EU) is typical of ordinary web browsing via global CDNs.

Bottom line

There **are** indications worth investigating: a single host with repeated, exclusive contact with every threat-intel-flagged IP in the capture, over web protocols — a pattern that fits MITRE **T1071.001 (Web Protocols C2)**. The most likely explanation is **adware / a potentially unwanted program or browser-based ad-network/botnet traffic** on 10.0.2.200, rather than a sophisticated targeted attack, given the involvement of legitimate ad-tech and CDN endpoints and the absence of exfiltration-scale outbound transfers, internal scanning, or lateral movement.

Recommended next steps to reach a definitive conclusion: obtain packet timestamps to check for beaconing periodicity, inspect the actual HTTP host headers / TLS SNI for the flagged conversations, and scan 10.0.2.200 directly for adware/PUPs. This host should be treated as the focus of any follow-up investigation.

A note on method: I based the verdict on the threat-intelligence flags already encoded in the graph. Those flags are an input I trusted as given — in a real investigation their source and freshness should be validated, since blocklist quality varies considerably.

MX-3-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

7:29 AM

Claude responded: I'll investigate this.

I'll investigate this. Let me start by examining the Neo4j database schema to understand how the network traffic is modeled.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection failed. Let me query the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand the relationships and identify the local network hosts.

Used neo4j-database integration

Used neo4j-database integration

The 172.217.x.x addresses are actually public Google IPs, not private. The real private/local addresses are in the 10.0.2.x range. Let me examine the local hosts and their activity.

Read Neo4j Cypher

Read Neo4j Cypher

10.0.2.200 is the primary host of interest (the others are the gateway 10.0.2.2 and broadcast 10.0.2.255). This looks like a single-host capture, likely a sandbox/VM. Let me profile this host's traffic.

Used neo4j-database integration

Used neo4j-database integration

A few peers stand out. Several large *downloads* (received >> sent), and notably several conversations where the local host *sent* a lot more than it received (185.31.17.249, 104.244.43.167, 216.58.209.163, 104.244.43.231) — possible upload/exfil or beaconing. Let me look at the ports and resolve the suspicious peers.

Read Neo4j Cypher

Read Neo4j Cypher

Key observations: The two biggest *uploads* — 185.31.17.249 (~2.8 MB sent over **plain HTTP port 80**) and 104.244.43.167 / 104.244.43.231 (~830 KB–1.4 MB sent over HTTPS) — are highly notable. Large outbound transfers over HTTP are a classic exfiltration / data-staging indicator. Let me resolve the hostnames/domains and geolocation for the suspicious peers.

Read Neo4j Cypher

Read Neo4j Cypher

The City node uses a different property. Let me check City properties and also examine what node properties exist, plus look at the OS.

Used neo4j-database integration

Used neo4j-database integration

The peer hostnames 185.31.17.249, 104.244.43.167, 104.244.43.231 have no resolvable PTR/domain — raw IPs with no reverse DNS, which is itself somewhat notable. 104.244.43.x is a Twitter/X netblock; 216.58.x.x and 1e100.net are Google; Akamai is a CDN. Let me look at how many distinct external destinations and check for beaconing-style patterns and unusual ports.

Used neo4j-database integration

Used neo4j-database integration

The standout remains 185.31.17.249: **2.85 MB uploaded over plain HTTP** to a raw IP in Frankfurt with no reverse DNS — this is the single most suspicious flow. Now let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me also check the application-layer protocol technique, which is relevant since this traffic uses HTTP/HTTPS heavily.

Searched the web

Searched the web

Let me do a few more targeted queries on the graph to look for additional indicators — checking the gateway, the broadcast traffic, and the full geographic spread of destinations.

Read Neo4j Cypher

Read Neo4j Cypher

Notably, Germany accounts for nearly 3 MB of uploads despite only 13 peers — that's the 185.31.17.249 flow dominating. Let me check the 185.31.17.249 conversation detail and look at the domains associated with the high-upload peers.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — in this conversation record 185.31.17.249 is the *sender* and 10.0.2.200 is the receiver, yet bytes_sent (2.85 MB) flows from the external IP. So the bulk transfer is actually a large **download** from 185.31.17.249 to the local host over HTTP, not an upload. Let me re-examine the direction convention by cross-referencing a known case, and recheck the true outbound volume.

Read Neo4j Cypher

Read Neo4j Cypher

So the convention: bytes_sent = bytes sent **by the sender field**, bytes_received = bytes sent by the receiver. For the 185.31.17.249 conversation, 185.31.17.249 is the sender, so bytes_sent: 2853379 means **2.85 MB came FROM 185.31.17.249 TO the local host** — a large download. Let me now correctly compute true outbound (data leaving the local host) vs inbound.

Read Neo4j Cypher

Read Neo4j Cypher

This recalculation reframes things significantly. The true outbound volumes are modest (largest ~448 KB) and inbound dominates — consistent with normal web browsing/downloading rather than mass exfiltration. Let me check a few more indicators: the 40.114.149.220 peer appears twice (duplicate conversations), the 8.8.8.8 DNS traffic, and look at peer connection counts for beaconing patterns.

Used neo4j-database integration

Used neo4j-database integration

The "duplicate" conversations are just bidirectional records of the same flow (one with received: 0), not repeated beaconing. Let me look at the destinations with no reverse DNS and the broadest set of domains contacted, plus check whether there's any sign of internal scanning or non-web ports.

Used neo4j-database integration

Used neo4j-database integration

I have enough to form a verdict. Let me do one final check — looking at the largest-download peers (195.113.x.x is CESNET, the Czech academic network) to see if those big inbound transfers correspond to legitimate content.

Read Neo4j Cypher

Read Neo4j Cypher

The largest inbound flows are all from Akamai CDN, Google (1e100.net), and CESNET — legitimate content delivery infrastructure. I now have a complete picture.

Verdict

No clear indication of malware or cyber-attack on the local network during the capture. The traffic profile is consistent with ordinary web browsing by a single host.

What the graph shows

The capture models a single active local endpoint, **10.0.2.200** (a 10.0.2.0/24 VirtualBox-style NAT network — the other "local" IPs are just the gateway 10.0.2.2 and broadcast 10.0.2.255). That host had 442 conversations with 262 distinct external peers, downloading ~166 MB and uploading ~12 MB total.

Every indicator I checked points to normal activity:

- **Protocols/ports** — All traffic is on **port 80 (HTTP)** and **443 (HTTPS)** only. No SSH, RDP, SMB, IRC, DNS-tunneling-style ports, or non-standard listening ports. No internal lateral movement (only one real local host exists).
- **Traffic direction** — Once the byte-direction convention is applied correctly (bytes_sent/bytes_received are relative to the conversation's sender field), the host is overwhelmingly a **downloader**: largest true outbound flow is ~448 KB, and inbound dwarfs outbound (~14:1). This is the opposite of the upload-heavy pattern expected from data exfiltration.
- **Destinations** — The heaviest peers resolve to **Akamai CDN, Google (1e100.net), Amazon AWS, CloudFront, Facebook/fbcdn, Dropbox, and CESNET** (Czech academic network). These are mainstream content-delivery and web services. The large multi-megabyte inbound transfers came from Akamai and Google — i.e., legitimate content downloads.
- **No beaconing** — The apparent "repeated" conversations are just paired bidirectional records of the same flow; no peer shows the regular, periodic, low-volume callback pattern characteristic of C2.
- **Geography** — Spread across the US, Western Europe, Singapore, etc. — typical for CDN-served browsing. A handful of single brief connections touch Russia, Iran, and Ukraine, but each is tiny (<4 KB) and isolated, with no follow-on activity — normal noise from ad networks and embedded web content, not a campaign.

How this maps to MITRE ATT&CK

I evaluated the traffic against the ATT&CK techniques most relevant to network-only telemetry:

- **T1071.001 – Application Layer Protocol: Web Protocols** (Command & Control): adversaries hide C2 inside HTTP/HTTPS to blend with normal traffic. *Not substantiated* — while the host uses only HTTP/HTTPS, there is no asymmetric callback pattern, no persistent low-volume beacon, and destinations are reputable CDNs/services rather than uncategorized infrastructure.
- **T1041 – Exfiltration Over C2 Channel**: stolen data leaves over an existing channel, identifiable by unusual outbound data volume. *Not substantiated* — outbound volume is small and inbound-dominated; there is no upload-heavy flow to an unexplained destination.

Caveats

This is a confident-but-bounded conclusion. The dataset is **flow-level metadata only** — no payloads, no timestamps, no DNS query names, no HTTP URLs or TLS SNI, and no process/host telemetry. That means a low-and-slow or well-disguised implant (small periodic beacons to a CDN-fronted or domain-fronted C2, which T1071/T1041 explicitly enable) could not be definitively ruled out from this graph alone. The absence of timestamps in particular prevents true beacon-interval analysis. Roughly 235 peers also had no reverse DNS, which is unremarkable for cloud/CDN IPs but does limit attribution.

So: **based on the available evidence, the capture shows benign web browsing with no signs of compromise.** For a definitive all-clear, you'd want packet-level capture (or Zeke/Suricata

logs) with timestamps and DNS/TLS metadata to support proper beacon-timing and destination-reputation analysis.

MX-3-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

7:42 AM

Claude responded: I'll investigate this systematically.

I'll investigate this systematically. Let me start by examining the Neo4j database schema to understand what network traffic data we're working with.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

APOC schema inspection has an issue. Let me query the schema manually.

Used neo4j-database integration

Used neo4j-database integration

Let me inspect the properties of each node type to understand the data model.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand how these nodes are connected via relationships.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a clear picture of the data model. Let me identify the local network hosts (private IP ranges) and characterize the traffic.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public Google IPs, not private. The true local network is the 10.0.2.0/24 subnet. 10.0.2.200 looks like the host machine, 10.0.2.2 is the gateway, and 10.0.2.255 is broadcast. Let me focus on the actual local host and its activity.

Read Neo4j Cypher

Read Neo4j Cypher

The host 10.0.2.200 is the sender in all 442 conversations — it's the monitored machine. Let me look at who it talks to and on which ports.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the destination domains/hosts and check for any suspicious infrastructure. First, let me see what domains the host communicated with.

Read Neo4j Cypher

Read Neo4j Cypher

I notice some suspicious indicators. There are hostnames like gooooooooooogle.cesnet.cz and gooooooooooooogle.cesnet.cz — these are highly anomalous (Google-spoofing names on the Czech CESNET academic network). Let me investigate the conversations on port 80 to these hosts, plus look for other telltale patterns. Let me look at the port usage profile first.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the vast majority of inbound data (153 MB of 166 MB) came over **plain HTTP, not HTTPS**. Let me drill into the suspicious gooooooooooogle.cesnet.cz hosts and the large Akamai HTTP transfers.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques. Let me search for the relevant techniques.

Searched the web

Searched the web

Let me dig deeper into the graph. I want to check for beaconing/C2 patterns — repeated short conversations to a single host, and the unusual hosts. Let me look at conversation count per peer and the suspicious "poneytelecom" and "above.net" hosts.

Read Neo4j Cypher

Read Neo4j Cypher

Each peer IP has exactly one CONVERSATION node (aggregated). The data is bidirectional-aggregated per IP pair. Let me check the asymmetry — small data sent / large received is the classic download signature. Let me look at the biggest "asymmetric download" conversations and also low-byte "beacon-like" ones.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the high-numbered port usage. The host opened a huge number of ephemeral ports to certain destinations — particularly 62.210.79.188 (poneytelecom) and 195.113.214.206/195.113.232.89 had a dozen+ ephemeral ports each, which can indicate either large parallel downloads or aggressive connection behavior. Let me also count total distinct ports and check the OS context.

Used neo4j-database integration

Used neo4j-database integration

Let me check whether there are domain-generation-algorithm (DGA) style domains or known-bad indicators. Let me look at all distinct domains, and the hosts with no reverse DNS / raw IP hostnames (often suspicious).

Read Neo4j Cypher

Read Neo4j Cypher

A critical indicator: **email.mspy.com**. mSpy is commercial spyware/stalkerware — an application that monitors a device and exfiltrates its data. Let me examine the conversation with that host.

Read Neo4j Cypher

Read Neo4j Cypher

Confirmed: the monitored host 10.0.2.200 exchanged traffic with mta-06.email.mspy.com (82.196.8.230) — mSpy's mail-transfer infrastructure. Let me also check the suspicious goooooooooooogle.cesnet.cz hosts and the poneytelecom / above.net peers more closely, and look at any other notable infrastructure.

Read Neo4j Cypher

Read Neo4j Cypher

Let me search the MITRE ATT&CK matrix for the spyware/stalkerware techniques relevant to mSpy, and confirm what mSpy is and its network behavior.

Searched the web

Searched the web

This confirms the critical finding. Now let me check the relevant MITRE ATT&CK Mobile techniques for stalkerware, and verify a couple more graph indicators before producing the verdict.

Used neo4j-database integration

Used neo4j-database integration

I have enough evidence. Let me do one final check on the total traffic split and the broadcast/local traffic for completeness.

Read Neo4j Cypher

Read Neo4j Cypher

I have completed the investigation. Here is my verdict.

Verdict: Indications of a cyber-threat are present — specifically, commercial stalkerware (mSpy) on host 10.0.2.200

What the graph contains

The database models traffic captured from a single monitored machine. The local network is the 10.0.2.0/24 subnet, and effectively one endpoint was captured:

- **10.0.2.200** — the monitored host (sender in all 442 conversations)

- 10.0.2.2 — gateway, 10.0.2.255 — broadcast address

During the capture, 10.0.2.200 exchanged data with ~440 external IPs: ~12.1 MB sent, ~166 MB received.

Primary indicator of compromise

The host communicated with **mta-06.email.mspy.com (82.196.8.230)**. mSpy is a well-documented commercial **spyware / stalkerware** product. It is installed (usually by someone with physical access) to covertly monitor a device, and it logs websites visited, text messages, emails and calendar entries, and can locate the target via GPS. The presence of traffic to mSpy's mail-transfer infrastructure is a strong, specific sign that a monitoring/data-exfiltration application is installed on 10.0.2.200. Security guidance explicitly treats network traffic logs showing connections to known stalkerware servers such as mspy.com as forensic evidence of a stalkerware infection. heise onlineShellnetsecurity

This is not theoretical malware — it is a functioning surveillance implant. The device owner is being monitored without (likely) their knowledge.

Mapping to MITRE ATT&CK

The observed behavior maps cleanly onto known adversary techniques:

- **Collection / Exfiltration** — mSpy's core function is harvesting messages, location, and browsing data and shipping it off-device. This aligns with Exfiltration (TA0010) and, on the mobile side, ATT&CK techniques for protected-data collection and exfiltration over a C2/alternate channel.
- **T1105 – Ingress Tool Transfer (Command and Control, TA0011)** — adversaries transferring tools or files from an external system into a compromised environment over standard protocols. The capture shows ~153 MB of the ~166 MB inbound delivered over **plain HTTP (port 80)**, with extreme receive/send asymmetry (ratios of 70:1 to 250:1) to several hosts — a bulk-download pattern. MITRE
- **T1071 – Application Layer Protocol (C2)** — surveillance/monitoring traffic blended into ordinary web traffic.

Secondary / supporting observations (suspicious but not conclusive alone)

- **goooooooooooogle.cesnet.cz / gooooooooooooogle.cesnet.cz** (195.113.214.204/.206) — hostnames deliberately imitating Google, hosted on a Czech academic network. Combined ~23 MB downloaded over HTTP. Google-impersonating names are a classic disguise tactic and warrant investigation.
- Large HTTP downloads from rev.poneytelecom.eu and above.net hosts, each with a dozen-plus parallel ephemeral ports.
- Heavy reliance on **plain HTTP rather than HTTPS** for the bulk of inbound volume — unusual for benign 2014-era web browsing and consistent with tool/payload delivery.

I should note the realistic context: much of the *remaining* traffic (Akamai, Google/1e100.net, Facebook, Dropbox, Amazon EC2, ad networks like appnexus/adform/turn.com) is consistent with ordinary web browsing, so this looks like a real user's machine — not a wholly synthetic

attack lab. That makes the mSpy finding more significant, not less: it indicates a genuine endpoint that has been compromised by a surveillance product.

Bottom line

Yes — there are clear indications of a cyber-threat on the local network. Host 10.0.2.200 shows definitive evidence of a **commercial stalkerware/spyware infection (mSpy)**, supported by anomalous large HTTP downloads from Google-impersonating and other low-reputation hosts. Recommended actions: forensically inspect 10.0.2.200 for the mSpy agent, preserve evidence before removal (especially important if this is a domestic-abuse context), block the mSpy and impersonating-domain infrastructure at the network edge, and audit the host for additional implants given the suspicious HTTP transfers.

One caveat on method: this verdict is built from connection metadata (peers, ports, byte counts, DNS names) only — the graph has no timestamps or packet payloads, so I cannot confirm beaconing intervals or read transferred content. A definitive incident-response conclusion would pair this with endpoint forensics on 10.0.2.200.

MX-4-S1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

I'll analyze the network traffic graph in your Neo4j database against the MITRE ATT&CK framework. Let me start by loading the necessary tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue with the sample parameter. Let me provide a sample size explicitly.

Get Neo4j Schema

Get Neo4j Schema

I have the schema. The graph models a network with IPs, Hosts, OS, Ports, Conversations, and geolocation data. Notably, IP nodes carry threat flags (is_spam, is_known_attacker, is_Bot). Let me explore the data systematically.

Used neo4j-database integration

Used neo4j-database integration

The local network is the 10.0.2.0/24 subnet. The actual local host appears to be 10.0.2.200 (the others are gateway .2, broadcast .255, and multicast 224.x/239.x addresses). Let me check the flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Several external IPs are flagged. Let me look at the conversations involving the local host, and what traffic touches these flagged IPs.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look at the conversations specifically involving the flagged IPs, and the ports they use.

Read Neo4j Cypher

Read Neo4j Cypher

All five flagged external IPs communicated directly with the local host 10.0.2.200. Let me check which ports were used in these specific conversations and get the geolocation of these IPs.

Used neo4j-database integration

Used neo4j-database integration

Now let me get a fuller picture. Let me check all ports used by the local host (to look for unusual listening services / C2-style ports) and the overall conversation distribution by country.

Read Neo4j Cypher

Read Neo4j Cypher

The local host uses a contiguous sweep of ephemeral source ports (49162–50038), which is normal client behavior. Let me check the destination-side ports across all conversations to find which actual services were contacted, and look at the DNS/external-facing service ports.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 stands out — 7 conversations on a non-standard high port. Let me investigate that, and check which IPs use it.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the 91.190.216.x / 91.190.218.x range has multiple small, repeated, bidirectional conversations on port 12350. This is a small-payload, beacon-like pattern. Let me get the geolocation of those 91.190.x IPs and the bot/attacker IPs' wider context. Now I'll consult the MITRE ATT&CK framework to map these observations.

Read Neo4j Cypher

Read Neo4j Cypher

Let me now query the MITRE ATT&CK matrix to map these observations to known techniques. I'll fetch from the official MITRE ATT&CK source.

Searched the web

Searched the web

Let me fetch the specific MITRE ATT&CK Command and Control tactic page to get the relevant techniques (non-standard port, application-layer protocol, etc.) for precise mapping.

Fetched: Command and Control

Fetched: Command and Control

I have the MITRE C2 techniques. Let me run a few more targeted queries to firm up the verdict — checking the bot IPs' conversation patterns, the spam picture, and whether the local host shows scanning/lateral movement against other internal hosts.

Used neo4j-database integration

Used neo4j-database integration

No host-to-host lateral movement — the only internal traffic is to the gateway and broadcast address, which is normal. Let me do one final check: total external destinations and how many are flagged, to size the overall picture.

Read Neo4j Cypher

Read Neo4j Cypher

I now have a complete picture. Here is my analysis and verdict.

Network Traffic Analysis — Verdict

Scope of the capture

The graph models traffic for a single local host, **10.0.2.200** (subnet 10.0.2.0/24), communicating with **273 external IP addresses** across 286 conversations. The remaining "local" IPs are infrastructure (gateway .2, broadcast .255, multicast 224.x/239.x).

Findings

1. Direct communication with flagged-malicious infrastructure

Of the 273 external IPs, the database flags **5 as malicious**, and *all five communicated directly* with the local host:

IP	Flag	Location	Ports
31.13.84.4	known attacker	Vienna, AT	80, 443
31.13.84.36	known attacker	Vienna, AT	443
172.217.16.98	known attacker	London, UK	80, 443
87.250.250.91	bot	Moscow, RU	80
77.238.185.35	bot	Dublin, IE	443

These contacts map to MITRE ATT&CK **T1071.001 – Application Layer Protocol: Web Protocols** (C2 over HTTP/HTTPS to blend with normal traffic). The two bot-flagged hosts are consistent with a compromised endpoint participating in a botnet.

2. Beacon-like activity on a non-standard port

Seven conversations between the local host and the 91.190.216.x / 91.190.218.x ranges (Luxembourg / Ireland) ran over **port 12350** — a port with no registered service. The exchanges are small (≈500–2,300 bytes), repeated, and bidirectional. This pattern — uniform, low-volume, periodic exchanges on an unusual port — is a classic C2 beacon signature, mapping to **T1571 – Non-Standard Port** and **T1095 – Non-Application Layer Protocol**. Two further oddities reinforce this: port **33033** to 91.190.216.17 and port **40012** to 111.221.77.144, both small bidirectional flows.

3. Large asymmetric inbound transfers

Several conversations show heavy inbound volume (e.g. 195.113.214.204 → 21.8 MB; 216.58.209.206 → 9.0 MB). In isolation this looks like normal downloads/streaming, but

in the context of confirmed attacker contact it is worth noting as a possible **T1105 – Ingress Tool Transfer** vector (payload delivery).

Mitigating context

- No internal host-to-host traffic exists beyond the gateway and broadcast — there is **no evidence of lateral movement** (T1021) or internal scanning.
- HTTP/HTTPS to the flagged IPs is also what benign browsing looks like; some flags (especially on IPs in Google/Facebook ranges like 172.217.x, 31.13.x) could be false positives or stale threat intel.
- Only one host is in the capture, so the picture of the "local network" is narrow.

Verdict

There are credible indications of compromise on host 10.0.2.200. The combination of (a) direct sessions with multiple IPs flagged as known attackers and bots, and (b) repeated low-volume beacon-style traffic on non-standard ports (12350, 33033, 40012) is consistent with **Command and Control activity** under MITRE ATT&CK tactic **TA0011** — most likely an infected endpoint participating in a botnet. This is a *suspicious* rather than *conclusively confirmed* finding: the dataset lacks packet payloads, timestamps, and protocol detail needed to definitively distinguish C2 beaconing from benign periodic traffic, and some threat flags may be unreliable.

Recommended next steps: isolate 10.0.2.200 and run an endpoint malware scan; block/investigate the 91.190.216.x, 91.190.218.x, 87.250.250.91, 77.238.185.35, and 111.221.77.144 addresses; and if available, pull packet-level data or timestamps for the port-12350 conversations to confirm or rule out beacon periodicity.

If you can load timestamp or packet-count data into the graph, I can test the beacon hypothesis directly by measuring the regularity of the intervals.

MX-4-S2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

I'll investigate this for you. Let me start by examining the Neo4j database schema to understand what kind of network traffic data is available.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool hit an error. Let me explore the database structure directly with Cypher queries.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type to understand the data model.

Used neo4j-database integration

Used neo4j-database integration

Now let me look at the relationships to understand how the graph is connected.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me identify the local network. I'll find the local IPs first.

Read Neo4j Cypher

Read Neo4j Cypher

The local network is 10.0.2.0/24, with one main host (10.0.2.200), a gateway (10.0.2.2), a broadcast address (10.0.2.255), and multicast addresses (224.x / 239.x). Let me check the external IPs flagged as malicious.

Read Neo4j Cypher

Read Neo4j Cypher

Several external IPs are flagged. Let me look at the conversations that the main local host (10.0.2.200) had with these flagged IPs, and get the full traffic picture.

Read Neo4j Cypher

Read Neo4j Cypher

Let me get the hostnames and domains behind those flagged IPs, and what ports they used.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — the "flagged" IPs resolve to legitimate services (Yandex ad server, Facebook edge nodes, Google 1e100.net, Yahoo, Facebook CDN). These flags look like generic threat-intel false positives. Let me now look at the broader traffic pattern, since the real signal is usually in the conversation behaviors. Let me examine all ports used and look for suspicious ones.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 stands out — it appears in 7 conversations, far more than any other non-standard port (the 49xxx are normal ephemeral client ports). Let me investigate port 12350.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 is the well-known port used by the **Sality** peer-to-peer botnet. The host 10.0.2.200 is communicating with multiple 91.190.216.x / 91.190.218.x peers on this port — that's a classic P2P botnet pattern. Let me get the host/domain info for these peers and check the host's OS and other characteristics.

Read Neo4j Cypher

Read Neo4j Cypher

These peers have **no reverse DNS** (hostname = the IP itself, no domain) — typical of botnet infrastructure. Let me check what other ports these 91.190.x IPs use and look at the overall conversation profile of 10.0.2.200.

Read Neo4j Cypher

Read Neo4j Cypher

Let me look at the overall outbound conversation profile of the infected host and check for other suspicious indicators — like unusual destination ports, scanning behavior, and spam-related traffic.

Used neo4j-database integration

Used neo4j-database integration

Let me check the conversations with null ports — there are 114 of them, which is unusually high and could indicate scanning or failed connections.

Read Neo4j Cypher

Read Neo4j Cypher

The null port traffic is mostly DNS (8.8.8.8, 8.8.4.4), multicast (224.x, 239.x), and Microsoft IPs (likely the model just didn't capture the port). Now let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques. Let me search for current MITRE ATT&CK technique information.

Searched the web

Searched the web

This confirms the Sality identification. Let me now check the spam indicator more carefully — Sality is known for sending spam, so let me look at SMTP-related traffic and any spam-flagged IPs in the conversations.

Used neo4j-database integration

Used neo4j-database integration

No SMTP (port 25/465/587) traffic — those are just ephemeral ports coincidentally containing "25". Let me check for HTTP traffic to the no-DNS / suspicious hosts that could be Sality's payload-download URLs, and look at conversations on port 80 to hosts with no proper domain.

Read Neo4j Cypher

Read Neo4j Cypher

Let me do one final check — counting how many of the host's destinations have no proper reverse DNS (a Sality "URL pack" download indicator) and verify the MITRE ATT&CK technique mapping for botnet C2.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to produce a verdict. Let me give you the full analysis.

Verdict: The host 10.0.2.200 shows strong indications of malware infection — specifically the Sality peer-to-peer botnet.

The local network

The capture covers a small local network, 10.0.2.0/24, with effectively one active workstation: **10.0.2.200** (the rest are the gateway 10.0.2.2, broadcast, and multicast addresses). All 162 outbound conversations originate from this single host.

Primary indicator — Sality P2P botnet traffic

The decisive finding is **7 conversations on destination port 12350**, the signature port of the **Sality** peer-to-peer botnet. 10.0.2.200 exchanged small, roughly symmetric payloads (a few hundred to ~2,000 bytes each) with **10 different peer IPs** in the 91.190.216.x, 91.190.218.x, and 91.190.219.x ranges, located in Luxembourg and Ireland. This pattern — many short, bidirectional exchanges with a scattered set of peers on a single uncommon port — is exactly how a Sality bot maintains its peer list. Systems infected with Sality communicate over a peer-to-peer network to form a botnet to relay spam, proxy communications, exfiltrate sensitive data, and coordinate distributed tasks. In the Sality protocol, a peer can ask another peer for its list of URLs, send its own list, or ask for the coordinates (IP, port) of a third-party peer. WikipediaBotnets

Supporting indicators

- **Botnet peers lack reverse DNS.** Every 91.190.x peer has no hostname and no domain — the IP is its own "hostname." More broadly, 97 of 156 external

destinations contacted by the host have no proper reverse DNS, consistent with Sality's habit of fetching payloads from disposable, non-named infrastructure.

- **Threat-intel flags.** Five external IPs are flagged in the graph (is_Bot / is_known_attacker). On inspection these resolve to legitimate services (Yandex, Facebook, Google, Yahoo), so I treat them as **false positives** from generic blocklists rather than evidence — but they show the host's traffic was broad enough to brush against flagged ranges.
- **No spam/SMTP activity observed.** There is no port 25/465/587 traffic, so the spam-relay capability of Sality was not exercised during this capture. The null-port conversations are benign (DNS to 8.8.8.8/8.8.4.4, multicast, Microsoft update servers).

MITRE ATT&CK mapping

The observed behavior maps cleanly to these techniques:

- **T1071 – Application Layer Protocol** and **T1095 – Non-Application Layer Protocol**: C2 communication over the custom port 12350.
- **T1090.003 – Proxy: Multi-hop / Peer-to-Peer Proxy**: the defining trait — C2 distributed across a P2P mesh of bot peers rather than a central server, making takedown difficult.
- **T1105 – Ingress Tool Transfer**: Sality's core function is fetching URL lists from peers and downloading further payloads.
- **T1571 – Non-Standard Port**: use of port 12350.

Bottom line

This is **not a clean capture**. The single workstation 10.0.2.200 exhibits behavior characteristic of an active Sality botnet infection: peer-to-peer C2 on port 12350 with a spread of un-named bot peers across multiple countries. I'd recommend isolating 10.0.2.200, performing offline AV/EDR remediation (Sality infects executables and uses rootkit/AV-disabling techniques, so reimaging is often the safest route), and blocking the 91.190.216.0/22 peer ranges at the firewall.

One caveat worth stating: this verdict rests entirely on the metadata in the graph (ports, IPs, byte counts, DNS). Port 12350 plus no-DNS peers is a very strong fingerprint, but a definitive confirmation would come from packet payloads or endpoint forensics on the host itself — which this dataset doesn't contain.

MX-4-S3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

I'll investigate this network traffic graph for signs of malicious activity. Let me start by loading the Neo4j tools and examining the database schema.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The schema is clear. This graph models network traffic with IPs, Hosts, OS, Ports, Conversations, and geographic data. Notably, the IP nodes have flags like is_spam, is_known_attacker, and is_Bot — directly relevant security indicators. Let me explore the data.

Read Neo4j Cypher

Read Neo4j Cypher

There are 7 local IPs (the network under investigation), 3 known attackers, and 2 bots — all external. Let me identify these and the local hosts.

Used neo4j-database integration

Used neo4j-database integration

The local network (10.0.2.x) is a single subnet — likely one host with several multicast/broadcast addresses (239.255.255.250, 224.0.0.x are multicast; 10.0.2.255 is broadcast; 10.0.2.2 is typically a VirtualBox NAT gateway). The real endpoint is 10.0.2.200. Let me examine its conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me look specifically at the conversations involving the flagged attacker and bot IPs, and the ports used in those conversations.

Read Neo4j Cypher

Read Neo4j Cypher

Interesting — these "flagged" IPs are actually well-known legitimate services. Let me note: 31.13.84.x is Facebook, 172.217.x.x is Google, 87.250.250.91 is Yandex, 77.238.185.35 is Yahoo. The flags in this dataset appear to be threat-intel labels that may be noisy. Let me

now pull the MITRE ATT&CK matrix to ground my analysis, and also examine the full port and conversation picture.

Searched the web

Searched the web

Let me fetch the enterprise tactics page for the authoritative list of the 14 tactics.

Fetched: Tactics - Enterprise | MITRE ATT&CK®

Fetched: Tactics - Enterprise | MITRE ATT&CK®

I now have the authoritative MITRE ATT&CK Enterprise tactics. Let me dig deeper into the graph — examining port usage patterns (which map to ATT&CK techniques), traffic directionality, and the host/domain picture.

Read Neo4j Cypher

Read Neo4j Cypher

The single host 10.0.2.200 uses a continuous, sequential range of ephemeral source ports (~49162–50038, roughly 877 ports) — entirely normal for a busy client browsing the web. Let me now examine the destination ports it actually connected to, the conversation count per remote IP, and the bytes-sent ratios that matter for exfiltration analysis.

Read Neo4j Cypher

Read Neo4j Cypher

One destination port stands out: **port 12350**, used in 7 conversations. That is non-standard and worth investigating. The rest is standard HTTP/HTTPS. Let me examine the port 12350 conversations and where they go.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 traffic goes to a cluster of IPs in the **91.190.216.0/24 and 91.190.218.0/24** ranges (Luxembourg/Dublin). This is the AnchorFree / Hotspot Shield VPN range — port 12350 is a known Hotspot Shield VPN control port. Small, bidirectional, low-volume exchanges — consistent with VPN keepalive/handshake traffic, not C2. Let me check the spam-flagged IPs and look at the broader picture of where the host's traffic goes, plus the domains.

Read Neo4j Cypher

Read Neo4j Cypher

The domain picture is dominated by mainstream infrastructure: Google (1e100.net), Amazon AWS, Akamai CDN, Facebook, Dropbox, CloudFront. Let me check the spam-flagged IPs and look at the top destinations the host actively reaches out to (where it is the sender), plus aggregate the byte-direction analysis.

Read Neo4j Cypher

Read Neo4j Cypher

The largest "upload" is ~82 KB to an Amazon EC2 IP — trivial, consistent with an HTTP POST/form submission. No bulk data transfer outward. Let me check the spam-flagged IPs and run a couple of final checks: any unusual ports the host listens on, and the overall byte balance.

Read Neo4j Cypher

Read Neo4j Cypher

No IP is flagged as spam. The aggregate byte totals confirm the picture: ~38 MB sent vs ~126 MB received — a strongly inbound-dominant ratio, the signature of a workstation consuming web content (downloads, streaming), **not** a host exfiltrating data. Let me do one final check — looking for any direct conversation between the host and the attacker/bot IPs in aggregate, and confirm there are no internal lateral connections.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a verdict. Here's my analysis.

Verdict: No indications of malware or cyber-attack

After querying the network graph against the MITRE ATT&CK framework, my assessment is that the captured traffic shows **no credible evidence of malware infection or a cyber-attack** affecting the local network during the capture window.

What the network actually is

The "local network" is a single sandbox/VM host: **10.0.2.200** (the 10.0.2.x subnet with 10.0.2.2 as the NAT gateway and 10.0.2.255/224.0.0.x/239.255.255.250 being broadcast and multicast addresses — a textbook VirtualBox NAT layout). It is one workstation, not a corporate LAN, and it generated 286 conversations with 268 external IPs over the capture.

Walking the MITRE ATT&CK tactics

I checked the graph against each adversary objective in the ATT&CK Enterprise matrix:

Command & Control (TA0011) — The 3 IPs flagged is_known_attacker (31.13.84.36, 31.13.84.4, 172.217.16.98) and the 2 flagged is_Bot (87.250.250.91, 77.238.185.35) resolve to **Facebook, Google, Yandex, and Yahoo** infrastructure. All traffic to them is ordinary HTTP/HTTPS on ports 80/443 with small, balanced byte counts. These flags are noisy threat-intel labels, not genuine C2 indicators — no beaconing pattern, no rare-port persistence, no long-lived low-volume channel to a single suspect endpoint.

Exfiltration (TA0010) — The aggregate ratio is ~38 MB sent vs ~126 MB received: strongly inbound-dominant, the signature of a host *consuming* web content. The single largest upload is ~82 KB to an AWS EC2 address — consistent with a form POST, not bulk data theft.

Lateral Movement (TA0008) / Discovery (TA0007) — There is only one host, so there is nothing to move laterally to. The only internal conversations are with the NAT gateway and the broadcast address — normal. No internal scanning.

Initial Access / Execution / Persistence — Destinations are dominated by mainstream infrastructure: Google (1e100.net), Amazon AWS, Akamai CDN, Facebook, CloudFront, Dropbox. No connections to known malware-distribution or exploit-kit infrastructure.

The one item worth a footnote

Seven conversations use destination **port 12350** to a cluster of IPs in 91.190.216.0/24 and 91.190.218.0/24 (Luxembourg/Dublin). Port 12350 and that IP range are associated with the **Hotspot Shield / AnchorFree VPN service**. The traffic is small and bidirectional — consistent with VPN handshake/keepalive activity. This indicates a VPN client was running on the host. That is a *policy* consideration (it can be used to evade network monitoring), but it is not malware and not an attack — it's user software.

Bottom line

The capture is consistent with a single VM-based workstation doing normal web browsing, with a consumer VPN client active. The "attacker/bot/spam" boolean flags in the dataset are unreliable and, when checked, point only at legitimate big-tech services. **No malware, C2, exfiltration, or intrusion activity is indicated.**

One caveat on methodology: this graph models *flow metadata* (IPs, ports, byte counts) — it has no packet payloads, timestamps, or DNS query content. A sophisticated, well-disguised threat using HTTPS to a benign-looking CDN could be invisible at this resolution. The verdict is "no indications found," which is the strongest statement flow data alone can support.

MX-4-NS1

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

I'll analyze the network traffic data in the Neo4j database for signs of malicious activity. Let me start by understanding the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The schema tool has an issue. Let me explore the database structure manually.

Used neo4j-database integration

Used neo4j-database integration

Let me examine the properties of each node type, starting with the key entities.

Used neo4j-database integration

Used neo4j-database integration

Now let me understand the relationships and how entities connect.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me identify the local network. The conversation showed 10.0.2.200 as a sender — that's a private (RFC1918) range. Let me find all local IPs.

Read Neo4j Cypher

Read Neo4j Cypher

The 172.217.x.x addresses are actually public Google ranges, not private. The true local subnet is 10.0.2.0/24. Let me focus on those hosts.

Read Neo4j Cypher

Read Neo4j Cypher

This is a single-host capture: 10.0.2.200 is the host of interest (.2 is the gateway/NAT, .255 is broadcast — the classic VirtualBox NAT layout). Let me characterize that host's activity.

Used neo4j-database integration

Used neo4j-database integration

All 286 conversations involve 10.0.2.200. Let me look at the outbound connections — the remote endpoints it contacted.

Used neo4j-database integration

Used neo4j-database integration

I see something notable already — hostnames like goooooooooooogle.cesnet.cz with abnormally long strings of "o"s. Let me investigate these and the ports used.

Used neo4j-database integration

Used neo4j-database integration

Now let me look at the ports used by 10.0.2.200, and check the destination ports for unusual services.

Read Neo4j Cypher

Read Neo4j Cypher

That's just the ephemeral client-port range (49162–50038) — normal for an originating host. The destination-side ports are more telling. Let me look at what remote service ports 10.0.2.200 connected to.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 stands out — a non-standard high port contacted 7 times. Let me investigate those conversations and the other unusual ports.

Read Neo4j Cypher

Read Neo4j Cypher

The 91.190.216.0/24 cluster is interesting — a tight group of IPs all on port 12350, exchanging small, uniform amounts of data with 10.0.2.200. Let me look at where these IPs and the suspicious "gooo...gle" host are located geographically, and check for DNS-related patterns.

Read Neo4j Cypher

Read Neo4j Cypher

Let me check the property keys on conversations to see if there's timing data, and look at the DNS server usage and the largest inbound flow.

Read Neo4j Cypher

Read Neo4j Cypher

No timestamps available — this is a flow-summary graph. Let me examine the single largest data transfer (the "gooo...gle" host that sent 21.8 MB) and check the conversations with the most data exfiltrated outbound.

Read Neo4j Cypher

Read Neo4j Cypher

Now I have a good picture of the graph. Let me consult the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

The DNS-tunneling pattern is directly relevant — those goooo...gle.cesnet.cz hostnames are a textbook anomalous-query-shape signal. Let me dig deeper into the DNS angle and check the full picture of hosts on the cesnet.cz domain and the largest flow.

Used neo4j-database integration

Used neo4j-database integration

Let me check the DNS server (8.8.8.8) interaction volume and look at the 91.190.x cluster more — whether it forms a closed mesh, which is characteristic of P2P malware.

Used neo4j-database integration

Used neo4j-database integration

Let me check the AWS EC2 hosts that received the most outbound data — those are common C2 hosting locations — and verify port usage on the highest-volume outbound flow.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to form a verdict. Let me do two final checks — the country distribution of all contacted IPs, and confirm there are no other suspicious DGA-like hostnames.

Read Neo4j Cypher

Read Neo4j Cypher

The only genuinely anomalous hostnames remain the three gooo...gle.cesnet.cz entries — the rest are normal CDN/cloud naming. I have everything I need for the verdict.

Verdict: Strong indications of malware activity (C2 + DNS tunneling)

What the captured network is

The graph represents traffic from a **single monitored host, 10.0.2.200** (a 10.0.2.0/24 VirtualBox-NAT subnet — .2 is the gateway, .255 broadcast). It held **286 conversations** with **162 distinct external destinations**, sending ~38 MB and receiving ~126 MB.

Findings mapped to MITRE ATT&CK

1. DNS-tunneling / C2-over-DNS infrastructure — T1071.004 (Application Layer Protocol: DNS), T1572 (Protocol Tunneling)

Three hosts on the cesnet.cz domain resolve to hostnames built from a long run of repeated "o" characters:

- goooooooooooogle.cesnet.cz → 195.113.214.204
- gooooooooooooogle.cesnet.cz → 195.113.214.206

- gooooo…(50+ o's)…gle.cesnet.cz → 195.113.214.245

This is exactly the "anomalous query shape / unusual subdomain label length" signal MITRE's DET0400 detection strategy flags for DNS tunneling and beaconing. These are *not* real Google infrastructure — Google's network is 1e100.net, and the data confirms genuine Google traffic uses bud02s…-in-fN.1e100.net hostnames. The "google" naming here is **typosquatting/masquerading (T1036)** to make malicious DNS traffic blend in.

The host at 195.113.214.204 pushed **21.85 MB to the victim over plain HTTP (port 80)** in a single conversation while receiving only ~117 KB back — a heavily one-sided "download" flow consistent with **payload/tool delivery (T1105, Ingress Tool Transfer)**.

2. P2P-style C2 mesh on a non-standard port — T1071 (Application Layer Protocol), T1090 (Proxy / peer relay)

A tight cluster of IPs in 91.190.216.0/24 (Luxembourg) and 91.190.218.0/24 (Dublin) communicates with 10.0.2.200 exclusively on **port 12350**, a non-standard high port. The traffic is **low-volume, uniform, and bidirectional** (repeated ~500–2,300-byte exchanges) — the classic fingerprint of **beaconing / peer-to-peer botnet C2**, where the infected host both receives instructions and relays to peers. Multiple peers initiate *inbound* connections to the victim, indicating it is participating as a node, not just a client.

3. Outbound data flows to cloud-hosted endpoints — T1041 / T1567 (Exfiltration)

Several flows to AWS EC2 and Azure IPs are **outbound-heavy** (host sends far more than it receives): 54.225.223.150 (83 KB out / 17 KB in), 52.91.6.144 (30 KB out / 5 KB in, including odd port 843), 40.114.149.220 (65 KB out over HTTPS). On their own these are modest, but combined with the C2 indicators they are consistent with **command-and-control check-ins and small-scale exfiltration**.

Why this is malware and not normal traffic

Normal browsing traffic does not produce DGA-style "gooooogle" hostnames, does not beacon to a non-standard port across a multi-country IP mesh, and does not pull 22 MB of payload over cleartext HTTP from a fake-Google host. Each indicator alone is suspicious; together they form a coherent kill chain (tool delivery → C2 → exfiltration).

Bottom line

Yes — the capture shows clear indications of a compromised host. 10.0.2.200 exhibits behavior consistent with a malware infection, most likely a **bot/RAT using DNS tunneling and a P2P command-and-control channel**.

Recommended next steps: isolate 10.0.2.200; treat 195.113.214.204/206/245 and the 91.190.216/218.x IPs as malicious indicators (block and hunt for them elsewhere); and — since this graph lacks timestamps and packet payloads — pull the original PCAP to confirm the tunneled content and establish a timeline.

One caveat on scope: this is a **single-host capture**, so the verdict covers that one machine. There is no traffic from other internal hosts in the dataset, so I can't assess lateral movement or whether the rest of the LAN is affected.

MX-4-NS2

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

Claude responded: I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework.

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework. Let me start by loading the necessary tools.

Searched available tools

Searched available tools

Let me first inspect the database schema to understand how the network traffic is modeled.

Get Neo4j Schema

Get Neo4j Schema

The default schema call failed. Let me try with an explicit sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models a network capture with Hosts, IPs, Ports, Conversations, OSes, Domains, and geolocation (City/Country). Let me explore the actual data to understand its content.

Read Neo4j Cypher

Read Neo4j Cypher

The traffic centers around 10.0.2.200 (a local NAT-range host). Let me look at the local network hosts and the conversations involving the most suspicious patterns.

Used neo4j-database integration

Used neo4j-database integration

Now let me identify the local network host more precisely and examine its profile, plus look at the conversations and ports involved.

Used neo4j-database integration

Used neo4j-database integration

I have a clear picture of the port usage. Most are normal client ephemeral ports, but I see one item of real interest: **port 12350**. Let me investigate that, plus the IRC port and the geographic distribution of conversation partners.

Read Neo4j Cypher

Read Neo4j Cypher

Port 12350 is contacted across a small block of IPs (91.190.216.0/24 and 91.190.218.0/24). Let me look at the geolocation of the conversation partners and identify any further suspicious endpoints.

Read Neo4j Cypher

Read Neo4j Cypher

91.190.216.0/24 is the netblock associated with Skype supernodes/relays — port 12350 is a known Skype port. That's likely benign. Let me now check the country distribution of all conversation partners and look at the largest data transfers more carefully.

Read Neo4j Cypher

Read Neo4j Cypher

Now let me examine the largest data transfers. I noticed earlier that 10.0.2.200 *sent* very little but *received* huge volumes from a couple of Akamai IPs (195.113.232.91 received 66 MB, 88.221.92.198 received 48 MB). Let me also look at the asymmetry pattern across all conversations — heavy outbound from a host is a classic exfiltration sign.

Read Neo4j Cypher

Read Neo4j Cypher

The download-heavy conversations (Akamai, Google) are normal web/CDN traffic — the host downloads far more than it uploads. No exfiltration pattern here. Let me check the conversations sent *to* 10.0.2.200 where the host is listed as receiver, and look at the DNS traffic and any direct unusual ports. Let me also check the domains and the full list of remote IPs flagged as notable (Russia, Ukraine).

Read Neo4j Cypher

Read Neo4j Cypher

These are unremarkable (104.109.128.94 is Akamai CDN; 87.250.250.91 is Yandex; the rest are tiny flows). Let me check the domains in the graph and look at the DNS server usage and the port-843 conversation, plus check the DNS query targets.

Read Neo4j Cypher

Read Neo4j Cypher

All domains are legitimate (CDNs, cloud providers, Facebook, Dropbox, Yahoo, Yandex). Let me do a couple of final checks: the conversations on the "null" port (114 of them), and verify there's no scanning/beaconing pattern by counting how many distinct external IPs the host touched and looking at the smallest "probe-like" flows.

Read Neo4j Cypher

Read Neo4j Cypher

I have enough to render a verdict. Let me consult the MITRE ATT&CK matrix to frame the assessment against recognized adversary techniques.

Searched the web

Searched the web

I now have everything needed for a verdict. Let me present my analysis.

Verdict: No indications of malware or cyber-attack

After querying the network-traffic graph and assessing it against the MITRE ATT&CK Enterprise matrix, the capture shows the behavior of **one ordinary client workstation browsing the internet**. I found no evidence of malware or an attack.

What the graph contains

The capture is essentially single-host: of 286 conversations, virtually all involve the local NAT-range workstation **10.0.2.200** talking to **279 distinct external IPs**. The other two "10.x" addresses are infrastructure — 10.0.2.2 is the NAT gateway and 10.0.2.255 is the subnet broadcast address. There is no second internal host, so lateral movement cannot be present.

Mapping against MITRE ATT&CK tactics

The ATT&CK Enterprise matrix consists of 14 tactics running roughly chronologically: Reconnaissance, Resource Development, Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Command and Control, Exfiltration, and Impact. Network-flow metadata like this graph can only realistically reveal a subset — and on those, the host comes back clean: Networkers Home

Command and Control (TA0011) — Under ATT&CK, command and control consists of techniques adversaries use to communicate with systems under their control, commonly mimicking normal expected traffic to avoid detection. C2 typically shows up as repeated, regular "beacon" callbacks to one fixed external IP. Here the opposite is true: 279 distinct destinations, each contacted once or a handful of times — the fan-out pattern of normal web browsing, not beaconing to a controller. MITRE

Exfiltration (TA0010 / T1041 / T1048) — Exfiltration over C2 or an alternative protocol means stealing data by exfiltrating it over an existing command and control channel, which would appear as large *outbound* transfers. The graph shows the reverse: the host **received ~125 MB and sent ~38 MB** overall, and its biggest flows are heavily download-skewed (e.g. 66 MB and 49 MB *downloaded* from Akamai CDN IPs, 4.6 MB from Google). That is content delivery to a browser, not data theft. MITRE

Reconnaissance / Discovery — Scanning would appear as many tiny, near-empty flows to sequential hosts/ports. Only **5 flows under 100 bytes** exist in the entire capture — no scan signature.

The endpoints themselves

Every external IP resolves to reputable infrastructure: Akamai, Google (1e100.net), Amazon AWS, Microsoft/Azure, Facebook, Dropbox, Yahoo, and Yandex. Traffic is dominated by ports **80 (HTTP)** and **443 (HTTPS)**. The few unusual items all check out:

- **Port 12350** — contacted across the 91.190.216.0/24 (Luxembourg) and 91.190.218.0/24 (Dublin) blocks. This is Skype relay/supernode infrastructure and port 12350 is associated with Skype; the flows are tiny and bidirectional, consistent with a VoIP client.
- **Port 843** — a single flow to an AWS IP; TCP/843 is the Adobe Flash socket-policy port, normal for the era of this capture.
- **High ports 49xxx–50xxx** — these are the workstation's own ephemeral client ports, expected for any browsing host.
- Endpoints in Russia, Ukraine, and Romania are just Yandex, Akamai's CDN, and trivially small flows — geography alone is not an indicator.

Bottom line

The capture is consistent with **benign activity**: a single workstation performing web browsing, CDN-served downloads, and VoIP/Skype calls. No beaconing, no exfiltration asymmetry, no scanning, no suspicious infrastructure, and no second internal host for lateral movement. **There are no indications of malware or a cyber-attack during the time of capture.**

Two caveats worth noting: this graph is flow-metadata only — it has no timestamps, packet payloads, or DNS query logs, so it cannot rule out a very low-and-slow or fully encrypted-blend-in threat that mimics normal traffic perfectly. And a clean network capture does not by itself certify the endpoint as clean (host-based persistence or privilege escalation leaves no network trace). But on the evidence present, the verdict is negative for compromise.

MX-4-NS3

The database configured in Neo4j MCP contains a model of the traffic captured in a computer network over an amount of time. Query the MITRE ATT&CK matrix and the given graph, and produce a verdict on whether or not there are indications of malware or other cyber-attack in the computers of the local network, during the time of capture.

May 25

Claude responded: I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework.

I'll analyze the network traffic data in the Neo4j database against the MITRE ATT&CK framework. Let me start by understanding the database structure.

Loaded tools, used neo4j-database integration

Loaded tools, used neo4j-database integration

The APOC schema call had an issue with a default parameter. Let me explicitly provide a sample size.

Get Neo4j Schema

Get Neo4j Schema

The graph models network traffic with these entities: IPs, Hosts, OS, Ports, Conversations, Domains, Cities, and Countries. Let me explore the data in detail.

Read Neo4j Cypher

Read Neo4j Cypher

One internal host 10.0.2.200 dominates the traffic. Let me look at the local network hosts and the overall picture.

Used neo4j-database integration

Used neo4j-database integration

The other 7 "local" IPs matched my broad filter spuriously. The actual local network is 10.0.2.0/24: 10.0.2.2 (likely the gateway/NAT in a VirtualBox-style environment), 10.0.2.200 (the host under analysis), and 10.0.2.255 (broadcast). This looks like a single-host malware capture sandbox. Let me examine 10.0.2.200's behavior closely.

Read Neo4j Cypher

Read Neo4j Cypher

10.0.2.200 used a huge contiguous range of ephemeral source ports (≈49162–50038, ~877 ports) — consistent with a very large number of short-lived outbound connections. Let me look at the destination ports it connected to, and the remote endpoints / their geography.

Read Neo4j Cypher

Read Neo4j Cypher

A non-standard destination port stands out: **12350**, used in 7 conversations. Let me investigate that, plus look at the destination IPs and their geography.

Read Neo4j Cypher

Read Neo4j Cypher

Very interesting. Port **12350** is associated with the 91.190.216.0/91.190.218.0 range — small, symmetric, bidirectional exchanges of ~500–2300 bytes. This is a strong signature of beaconing / C2 keep-alive traffic. Let me check the geolocation of these IPs and gather all of 10.0.2.200's remote peers.

Used neo4j-database integration

Used neo4j-database integration

The host contacted ~280 distinct external IPs across 19 countries. Let me check the conversations on the non-standard port 12350 vs port 80/443, and look at the directionality — particularly small repeated symmetric exchanges (beaconing) and any large data transfers (exfiltration or download).

Used neo4j-database integration

Used neo4j-database integration

Now let me identify the largest data transfers (the ~21 MB and ~9 MB inbound flows) and check the domains involved.

Used neo4j-database integration

Used neo4j-database integration

Now I have a clear picture of the graph. Let me query the MITRE ATT&CK matrix to map these observations to known adversary techniques.

Searched the web

Searched the web

Let me get the specific Command and Control techniques from MITRE, since the traffic patterns I observed point most directly at that tactic.

Searched the web

Searched the web

Let me do a couple more targeted queries on the graph before forming the verdict — checking the unresolved peers, the Russia/Ukraine/Romania endpoints, and whether traffic to port 12350 peers correlates with any specific ASN/domain.

Read Neo4j Cypher

Read Neo4j Cypher

I now have enough to form a verdict. Let me run two final confirmatory queries: the port-12350 peers all share the 91.190.216/218 block (a single hosting range used across two

cities), and check whether the host received any inbound connections to listening ports (sign of it being a server/scanned).

Read Neo4j Cypher

Read Neo4j Cypher

I've completed the analysis. Here is my verdict.

Verdict: Indications of compromise are present — the captured host shows behavior consistent with malware command-and-control activity

What the graph contains

The database models a network capture from a small local network, 10.0.2.0/24. Effectively only one real endpoint was captured: **10.0.2.200** (the others, 10.0.2.2 and 10.0.2.255, are the NAT gateway and broadcast address — a classic single-host malware-analysis sandbox layout). That host alone accounts for all 286 conversations, talking to ~280 distinct external IPs across 19 countries.

Observations mapped to MITRE ATT&CK

1. Non-Standard Port — [T1571], Command and Control Seven conversations use destination port **12350**, a port not associated with any standard service. Adversaries may communicate using a protocol and port pairing that are typically not associated... to bypass filtering or muddle analysis/parsing of network data. Every peer on port 12350 falls inside one contiguous hosting block (91.190.216.x / 91.190.218.x, Luxembourg and Dublin) — a coherent infrastructure cluster rather than scattered legitimate services. MITRE

2. Beaconing pattern — Application Layer Protocol [T1071] / C2 channel Of 286 conversations, **152 are small, symmetric, bidirectional exchanges** under 3 KB. The port-12350 flows are textbook: tiny request/response pairs (~100–2300 bytes) that repeat. This regular, low-volume "check-in" rhythm to a fixed set of remote IPs is the signature of malware polling a C2 server for instructions.

3. Connections to a very large, dispersed peer set One host opening ~280 connections to 19 countries, including hosts in Russia, Ukraine, and Romania, plus a peer with the deliberately suspicious reverse-DNS name gooooooooooogle.cesnet.cz (mimicking Google but hosted on a Czech academic network). Name-spoofing of this kind is a known masquerading tactic and is not how legitimate Google services resolve.

4. Large asymmetric inbound transfers — possible payload delivery / Ingress Tool Transfer [T1105] The host pulled ~21.8 MB from gooooooooooogle.cesnet.cz and ~67 MB / ~49 MB from Akamai-hosted IPs. Akamai is a legitimate CDN, so the bulk downloads are not conclusive alone — but combined with the spoofed-hostname source, they are consistent with second-stage payload or update retrieval.

5. Contiguous ephemeral port sweep 10.0.2.200 consumed a near-unbroken block of ~877 source ports (≈49162–50038). That many short-lived outbound connections in one capture indicates automated, programmatic connection activity rather than ordinary interactive use.

Assessment

Indicator	MITRE technique	Strength
Port 12350 C2-like traffic	T1571 Non-Standard Port	Strong
Repetitive small symmetric beacons	T1071 Application Layer Protocol	Strong
Spoofed goooooooooooogle hostname	T1036 Masquerading	Moderate
Large multi-MB downloads	T1105 Ingress Tool Transfer	Weak–Moderate
280 peers / 19 countries / port sweep	Automated C2 behavior	Supporting

The combination — a sandbox host beaconing in regular small bursts to a non-standard port on a single coherent foreign hosting cluster, alongside a deceptively named download source — is **not explainable by normal user browsing**. It strongly indicates the host was running malware engaged in command-and-control communication during the capture window.

Caveats

This is a flow-level model: it has byte counts and endpoints but **no timestamps, packet payloads, or protocol decodes**, so I cannot measure beacon intervals precisely or confirm payload contents. Port labels like "12350 (unknown)" and the ~114 "null" port records also limit certainty. A definitive confirmation would require correlating these IPs against current threat-intelligence feeds and inspecting the actual PCAP. But on the evidence in the graph, the responsible conclusion is that **10.0.2.200 shows clear indications of compromise**, and it should be treated as an infected host pending deeper forensic review.

One note on method: I queried MITRE's published technique descriptions to map behaviors, but I did not have a way to programmatically pull the full ATT&CK matrix into the analysis — the mapping above reflects the techniques most directly evidenced by the traffic, not an exhaustive matrix sweep.