COMP3010HK Security Operations and Incident Management 2025/2026

Coursework 1: Set Exercises - PCAP Analysis

Student ID: 10953435

Github repo: https://github.com/aka331/COMP3010HK/tree/main/CW1

Word count: 988

Section 1 – Introduction

This report presents a network traffic investigation based on the provided PCAP file and associated quiz questions. The objective is to identify the infected system, understand how the infection occurred, and determine the nature of the malware or attack involved.

The structure of this report is as follows:

Section No.	Content
Section 1	highlights the report objective;
Section 2	describes the tools and methodology used to analyse the PCAP and extract key indicators of compromise (IOCs);
Section 3	presents the main findings, drawing on the quiz answers and supporting evidence from the traffic analysis;
Section 4	concludes with prevention techniques, open challenges, and relevant references.

Table 1. Report Structure

Table of Content

Section 2 – Methodology

2.1 Wireshark

Firstly, Wireshark was used to investigate the PCAP. I filtered HTTP GET traffic to find the first malicious download and extracted the timestamp, domain, file name, and server headers. I exported the downloaded archive via Export Objects to confirm the embedded file. I then used DNS/TLS evidence and Conversations statistics to identify C2 IPs and map them to domains/Host headers. Lastly, I followed the SMTP TCP stream and decoded Base64 authentication data to obtain the required email and password details. Below are the expressions used for analyzing the traffic in the pcap file.

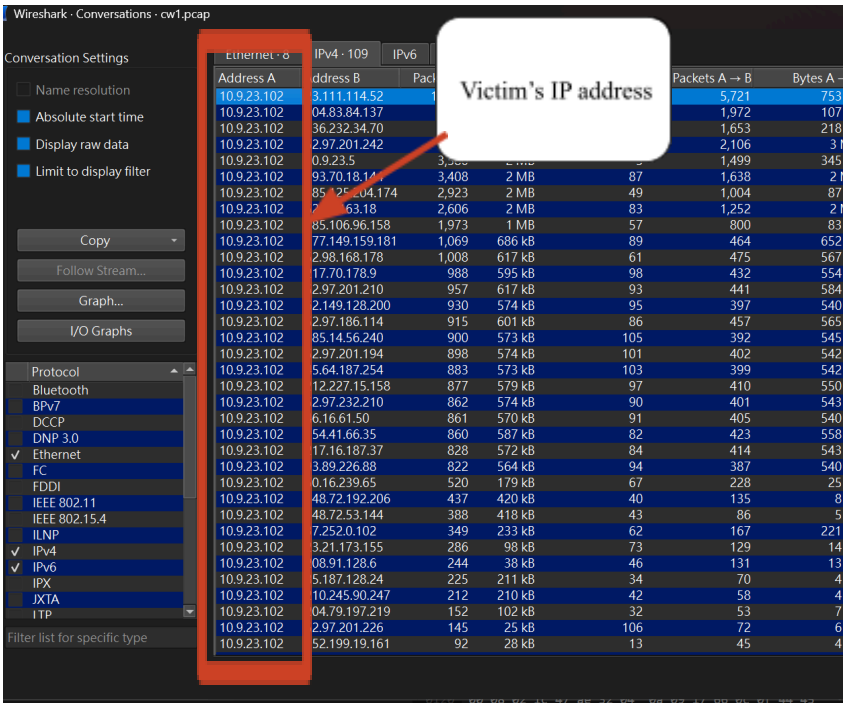
```
## Filtering only HTTP request traffic
http.request
## Filtering only HTTP GET requests
http.request.method == "GET"
## Filtering only the source IP from the victim and the HTTP POST traffic
ip.addr==10.9.23.102 && http.request.method=="POST"
## Filtering only the source IP from 10.9.23.102 and the DNS protocol traffic
ip.src==10.9.23.102 && dns
## Filtering only the IP from 185.106.96.158 (C2 server) and the HTTP traffic (port 80)
ip.addr == 185.106.96.158 && http
## Filtering only the IP from 185.125.204.174 (C2 server) and the port 8080 protocol traffic
ip.addr == 185.125.204.174 && tcp.port == 8080
## Filtering only the source IP from the victim, the DNS protocol traffic, and containing “api”
ip.addr==10.9.23.102 && dns && frame contains "api"
```

```
## Filtering only the source IP from the victim, the SMTP protocol traffic, and containing “FROM”
ip.addr==10.9.23.102 && smtp && frame contains "FROM"
```

Expression used in Wireshark

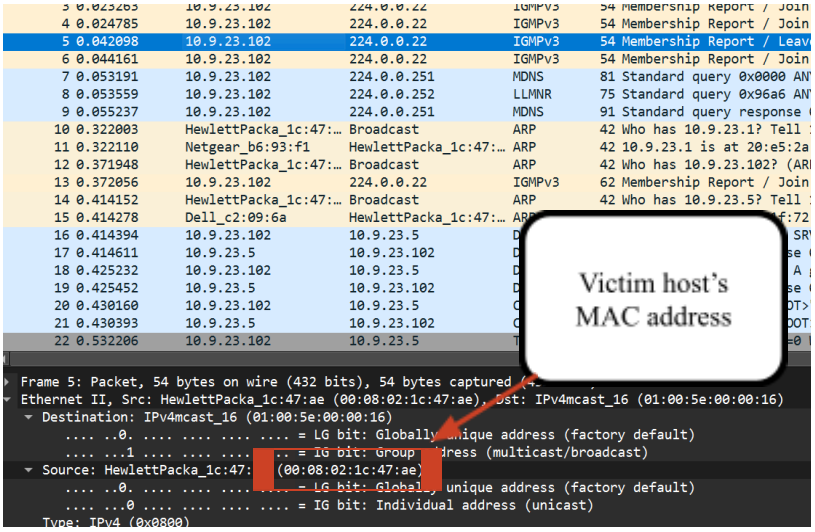
2.1.1 Identifying the Infected Host

The figure below shows that Address A(**10.9.23.102**) has a lot of conversations in descending order with other IP addresses (internal or external), which means the victim host is “**10.9.23.102**”



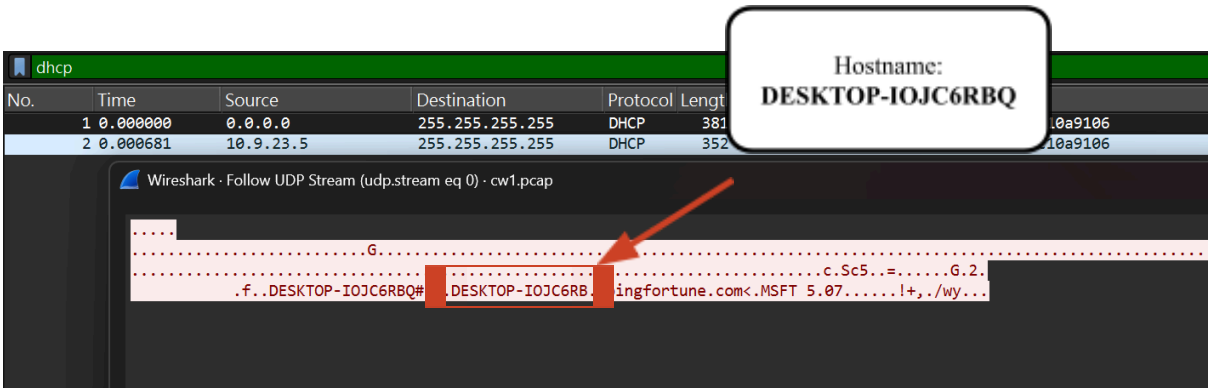
Conversation table in Wireshark

Select frame 5, where the source IP address(**10.9.23.102**), and review the MAC address(**00:08:02:1c:47:ae)** of the victim host.



Victim host MAC address

For hostname, run expression "**dhcp**", then right click "**Follow > UDP Stream**", the UDP stream shows the hostname is **DESKTOP-IOJC6RBQ**



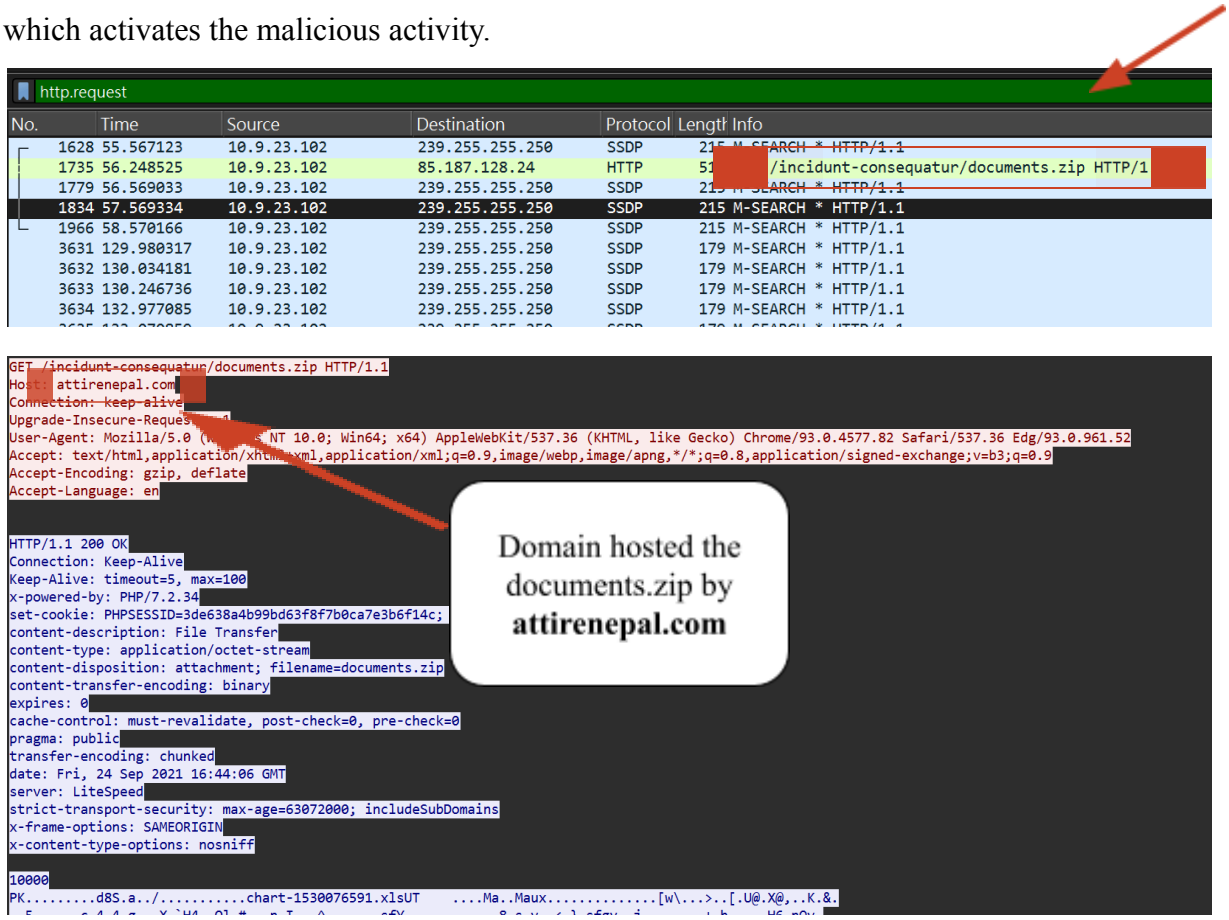
Victim hostname

The infected system information

Item	Information
IP address	10.9.23.102
MAC address	00:08:02:1c:47:ae
Hostname	DESKTOP-IOJC6RBQ

2.1.2 How the system got infected

Apply the HTTP request filter, which shows the victim, and download the **documents.zip**, which activates the malicious activity.



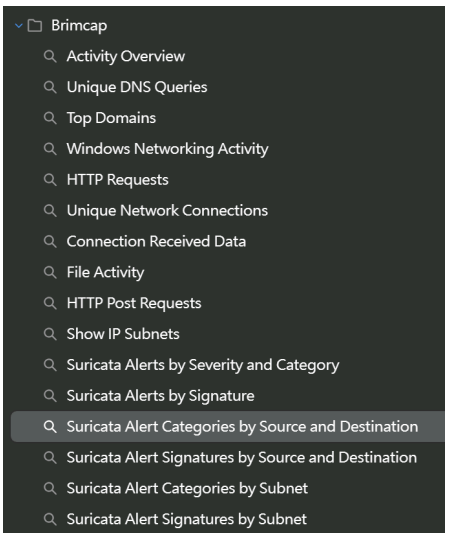
Malicious file from **attirenepal.com**

2.2 ZUI

Second, ZUI is a network forensic tool that provides a GUI interface to analyze and visualize large amounts of network traffic. It was utilized to analyze the CW1.pcap to investigate the network traffic that easily identifies the alert inside the PCAP file. It is similar to the SOC dashboard environment that allows us to filter out the event types of interest.

ZUI Screenshot 1. Activity overview

The preloaded query.json from brimcap on GitHub was used in the analysis of the pcap file. There is some information that can be viewed, such as activity overview, file activity, HTTP post requests, show IP subnets, Suricata Alerts by Severity and Category, Suricata Alerts by Signature, etc. Those preloaded queries are useful for analyzing malicious events.

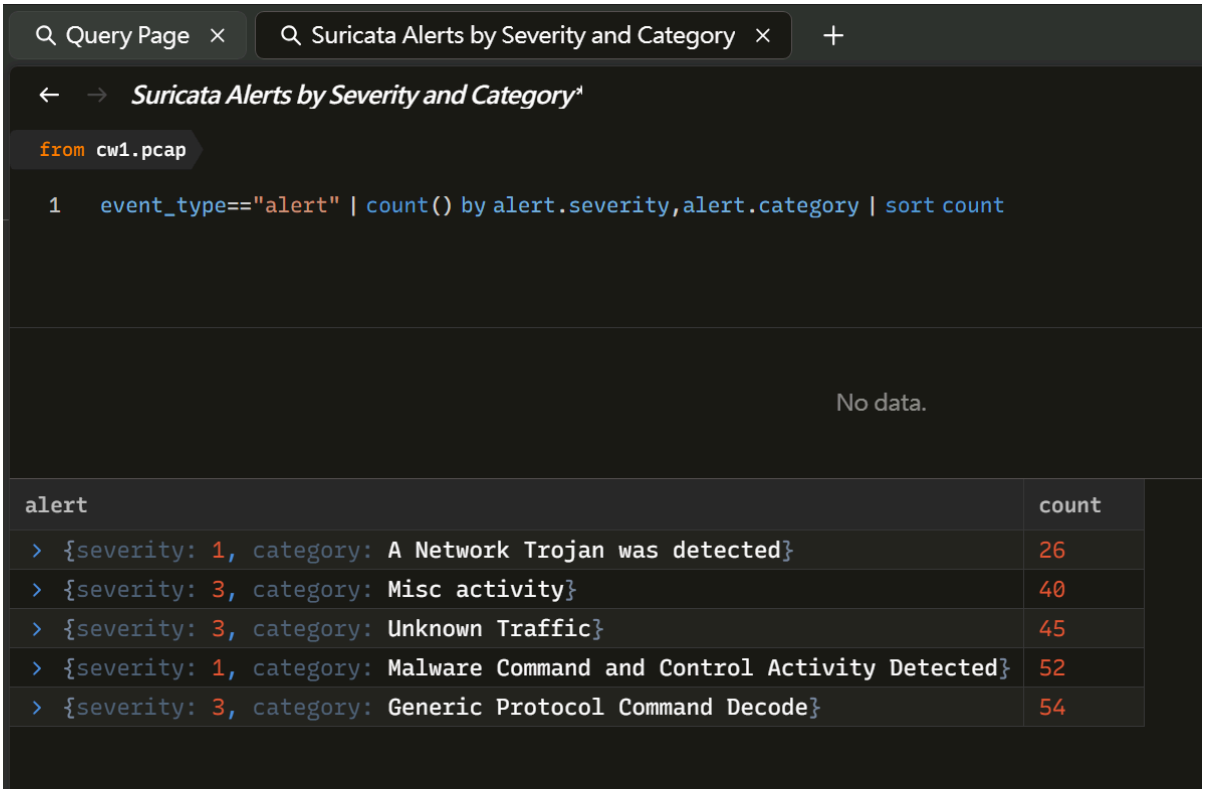


ZUI Screenshot 2. Preloaded Query

For alert triage, key attributes are those that can uniquely identify a rule, describe the relevant network session, and allow association with other logs. There are key attributes that can be used to analyze alert events, such as timestamp, src_ip, src_port, dest_ip, dest_port, proto, app_proto, alert.signature, alert.signature_id, alert.category, and alert.severity.

ZUI Screenshot 3. One of the JSON files in the alert events

From ZUI, we can identify the possible or potential attack that the victim is dealing with. The screenshot below shows that the attack could be a network trojan or malware activity.

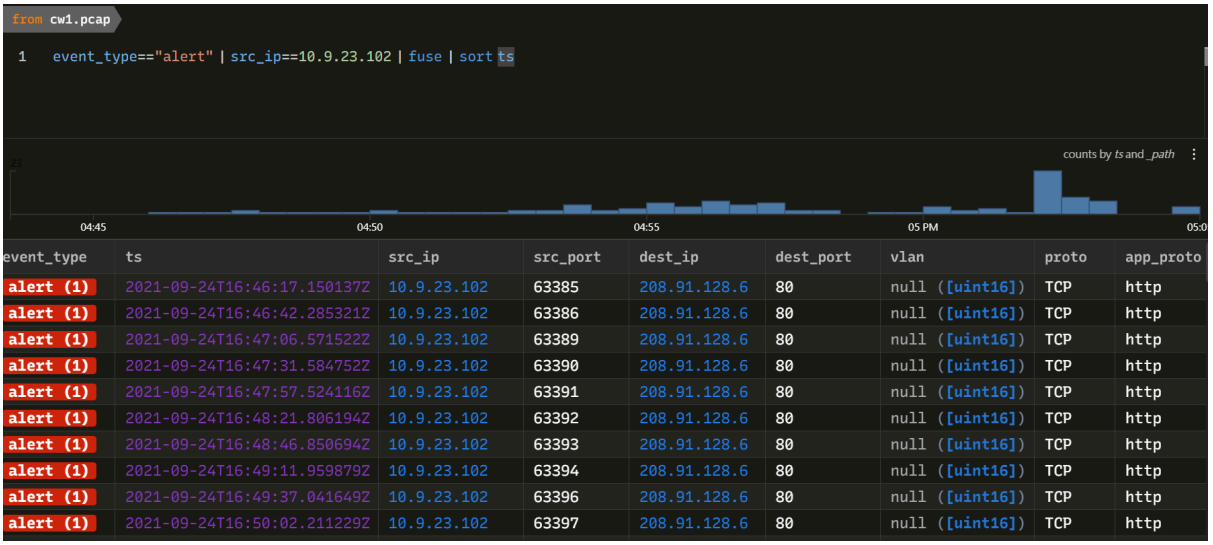


ZUI Screenshot 4. Suricata Alerts by Severity and Category

2.2.1 Infection Indicators

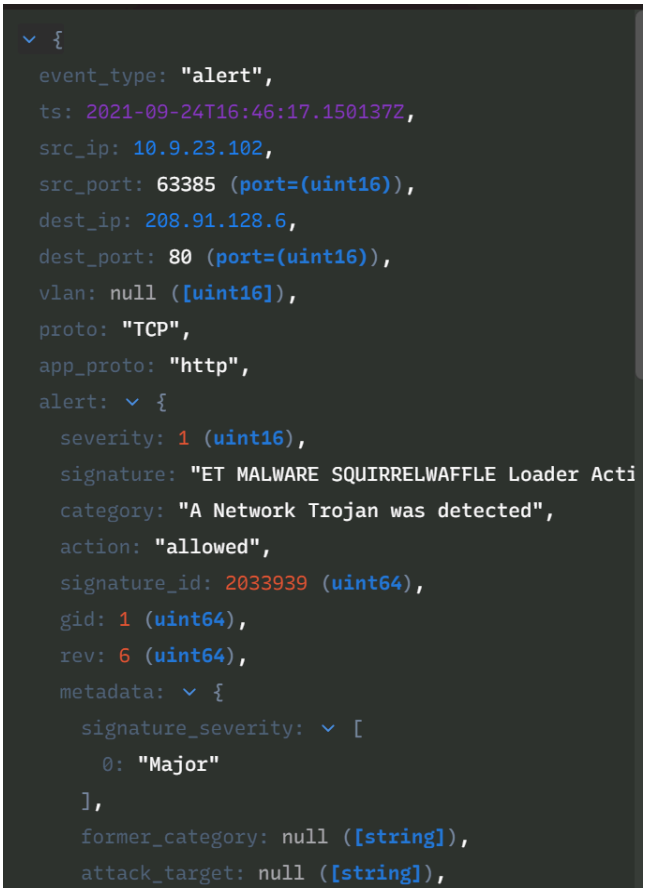
Firstly, ZUI was used as a primary analysis tool to identify infection indicators within the captured PCAP file. The malicious connection was started on

2021-09-24T16:46:17.150137Z



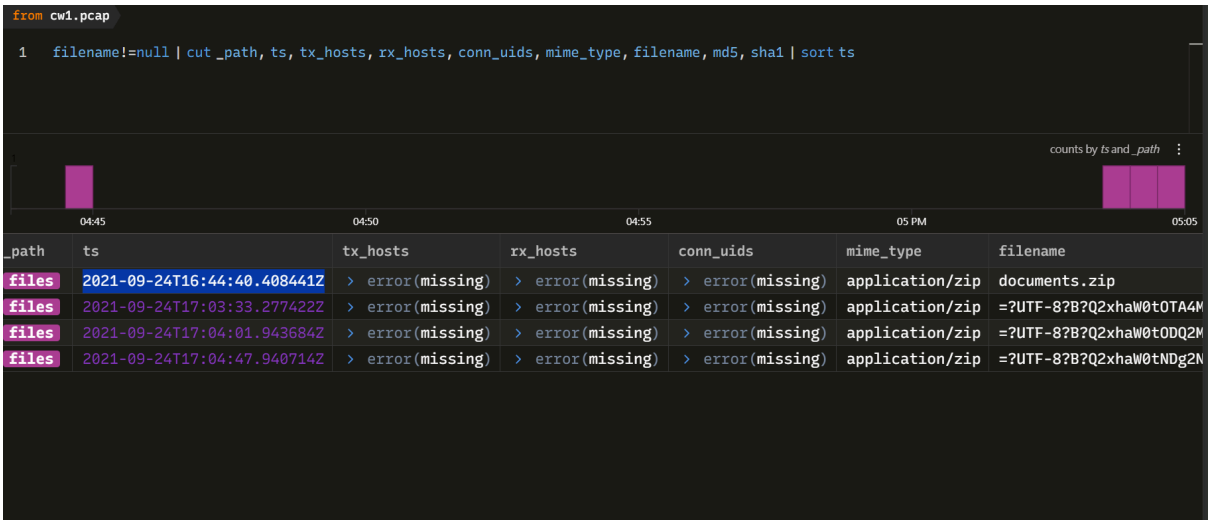
```
from cw1.pcap
1  event_type=="alert" | src_ip==10.9.23.102 | fuse | sort ts
```


event_type	ts	src_ip	src_port	dest_ip	dest_port	vlan	proto	app_proto
alert (1)	2021-09-24T16:46:17.150137Z	10.9.23.102	63385	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:46:42.285321Z	10.9.23.102	63386	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:47:06.571522Z	10.9.23.102	63389	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:47:31.584752Z	10.9.23.102	63390	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:47:57.524116Z	10.9.23.102	63391	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:48:21.806194Z	10.9.23.102	63392	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:48:46.850694Z	10.9.23.102	63393	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:49:11.959879Z	10.9.23.102	63394	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:49:37.041649Z	10.9.23.102	63396	208.91.128.6	80	null ([uint16])	TCP	http
alert (1)	2021-09-24T16:50:02.211229Z	10.9.23.102	63397	208.91.128.6	80	null ([uint16])	TCP	http

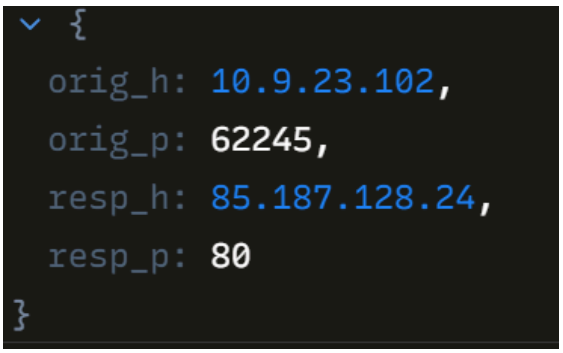


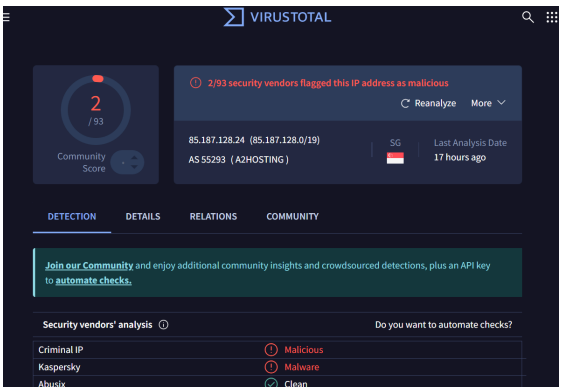

```
{
  event_type: "alert",
  ts: 2021-09-24T16:46:17.150137Z,
  src_ip: 10.9.23.102,
  src_port: 63385 (port=(uint16)),
  dest_ip: 208.91.128.6,
  dest_port: 80 (port=(uint16)),
  vlan: null ([uint16]),
  proto: "TCP",
  app_proto: "http",
  alert: {
    severity: 1 (uint16),
    signature: "ET MALWARE SQUIRRELWAFFLE Loader Acti
    category: "A Network Trojan was detected",
    action: "allowed",
    signature_id: 2033939 (uint64),
    gid: 1 (uint64),
    rev: 6 (uint64),
    metadata: {
      signature_severity: [
        0: "Major"
      ],
      former_category: null ([string]),
      attack_target: null ([string]),
```

Second, there has been a malicious file named "**documents.zip**" downloaded on **2021-09-24T16:44:40.408441Z**; the documents.zip classifier has been marked as a malicious file based on Virustotal analysis. (https://www.virustotal.com/gui/file/77229c744a0b1470afc7989a774cfe821386c11c0165e7e3fb5e9897a789a8cb)

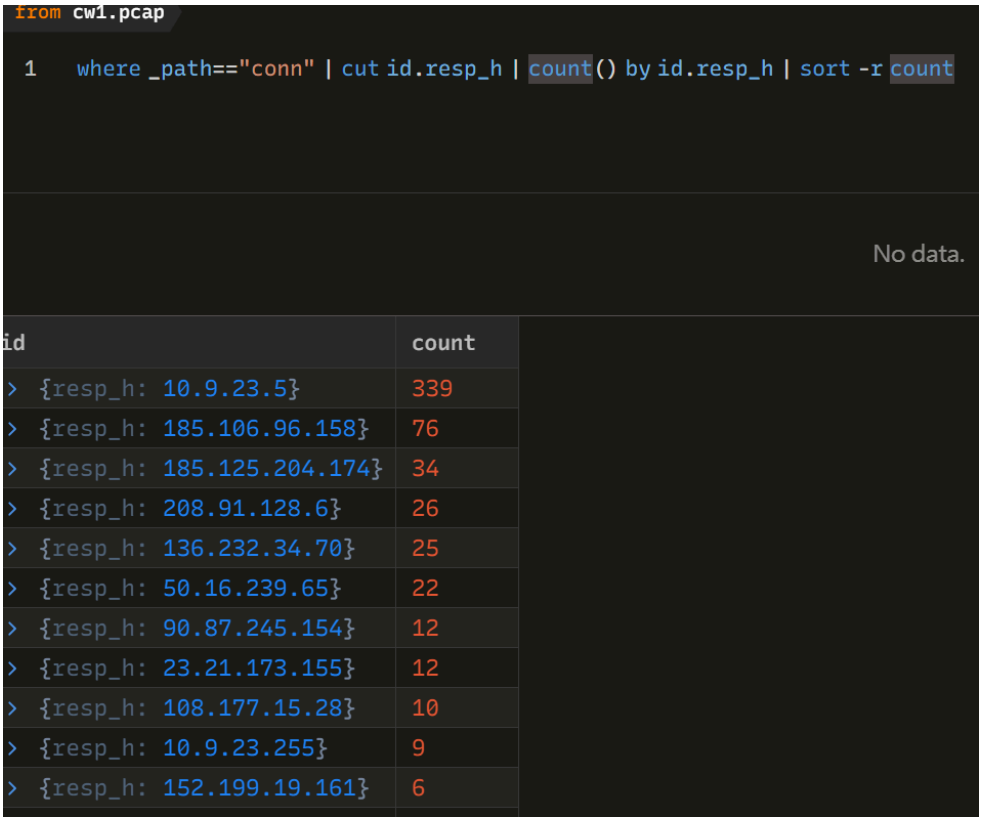


The file server is hosted at **85.187.128.24**, which is a malicious IP address. According to Virustotal analysis, this classifier has been flagged as a malicious IP address.





Third, to filter out what destination IP address communicates the most between the victim, run the zed language query "**where _path=="conn" | cut id.resp_h | count() by id.resp_h | sort -r count**", which is the result as below:



3 IP addresses request the most, which are 10.9.23.5, 185.106.96.158, and 185.125.204.174. Then submit those addresses to Virustotal. It found that **185.106.96.158** and **185.125.204.174** are C2 servers that allow attackers to remotely control infected systems, receive data from them, and issue malicious commands. (Virustotal Screenshot 1&2)

Fourth, when the malicious activity occurs, it has to analyze what DNS (Domain Name Server) is communicated after the victim opens the maliciouse excel file. Run the zed language query "**where _path=="dns" | where ts >= 2021-09-24T16:45:11Z | where ts <= 2021-09-24T17:45:35Z | cut ts, query | sort ts**" based on the hint.

ts	query
2021-09-24T16:45:11.44058Z	finejewels.com.au
2021-09-24T16:45:16.91361Z	self.events.data.microso
2021-09-24T16:45:20.26264Z	_ldap._tcp.default-first
2021-09-24T16:45:20.487123Z	thietbiagt.com
2021-09-24T16:45:25.457503Z	new.americold.com

After applying the Virustotal checking, **finejewels.com.au**, **thietbiagt.com**, and **new.americold.com** were malicious.

2.3 Virustotal

Virustotal is used to determine whether an external IP address, domain name, URL, suspicious file, or MD5 hash value is malicious. These evaluations may have come from the security community.

Virustotal Screenshot 1. Virustotal review (1)

Virustotal Screenshot 2. Virustotal review (2)

Section 3 – Results

After the investigation, we found that the victim's host was infected after downloading a ZIP file from a suspicious website. That ZIP contained a malicious Excel file. Shortly after, the host began beaconing to attacker-controlled infrastructure (C2) and later started sending large volumes of phishing/spam emails directly to many SMTP servers (port 25), including messages with a malicious ZIP attachment.

Initial infection & file transfer: At **2021-09-24 16:44:38** (Figure 1), the victim made an HTTP GET request to **attirenepal.com,** retrieving **documents.zip** (Figure 2&3).

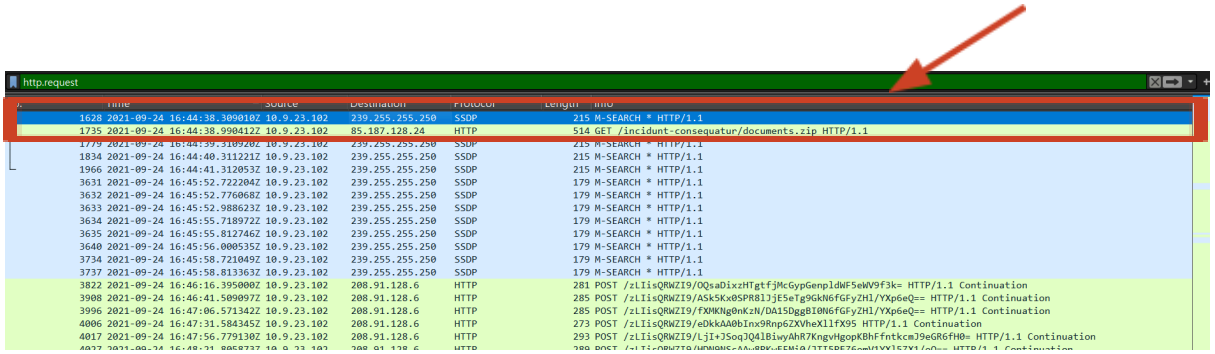


Figure 1. HTTP request

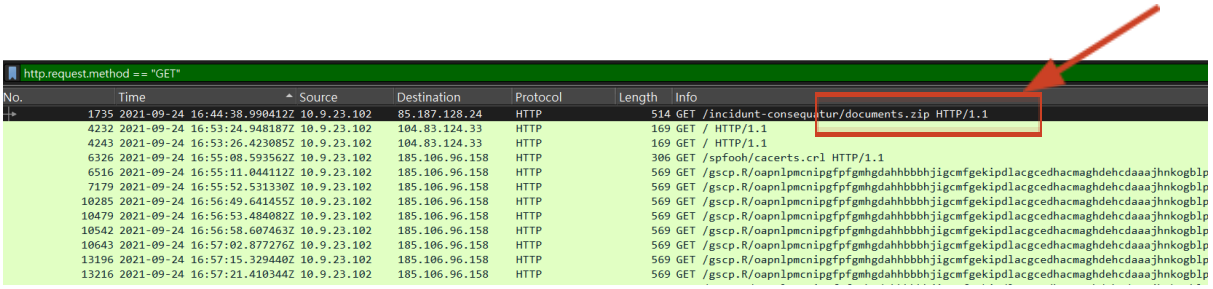


Figure 2. HTTP GET request that discovers document.zip

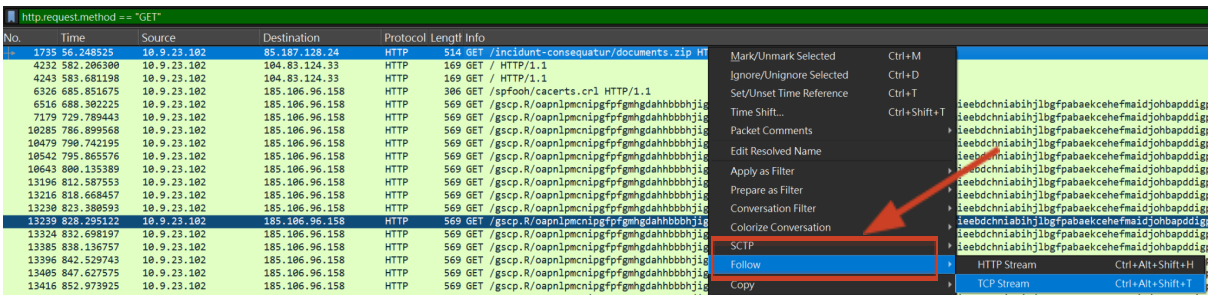


Figure 3. Select TCP Stream to view the dedicated traffic from GET /incidunt-consequatur/documents.zip HTTP/1.1

Exporting HTTP objects confirmed the archive contained **chart-1530076591.xls** (Figures 4 & 5). The malicious web server responded with Server: **LiteSpeed** with **PHP/7.2.34** (Figure 6), indicating the attacker-controlled infrastructure delivering the initial payload.

Figure 4. TCP Stream for /incidunt-consequatur/documents.zip

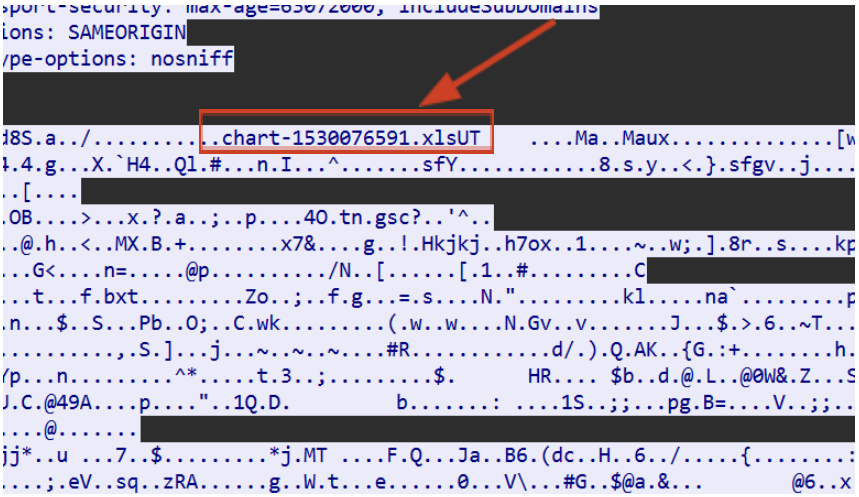


Figure 5. chart-1530076591.xls inside the stream

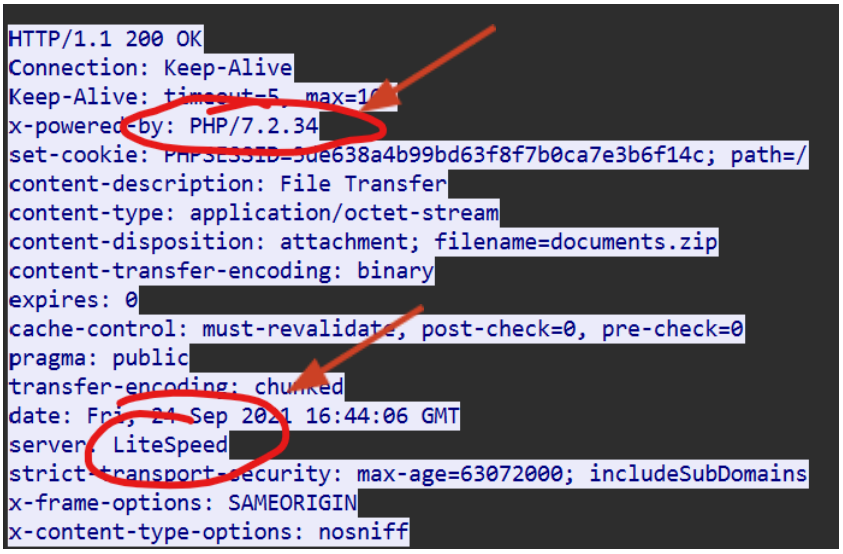


Figure 6. Web server and version

Additional download infrastructure: Within 16:45:11–16:45:30 UTC, additional domains contacted by the victim were **finejewels.com.au**, **thietbiagt.com**, and **new.americold.com,** as shown via DNS/TLS evidence (Figure 7). For the first domain, the TLS certificate issuer (CA) was **GoDaddy** (Figure 8).

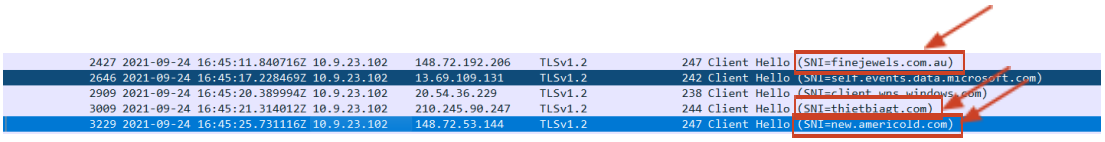


Figure 7. DNS traffic

Figure 8. CA issuer

Suspected C2 infrastructure consistent: Using Statistics → Conversations, two C2 candidate servers were identified: **185.125.204.174** (Port 8080) and **185.106.96.158** (Port 80) (Figures 9 & 10 & 11).

DNS analysis linked **185.106.96.158** (Port 80) to **survmeter.live** (Figure 12), and the Host/SNI value observed for this infrastructure was **ocsp.verisign.com** (Figure 13).

For **185.125.204.174**, HTTP POST traffic revealed an association with **securitybusinpuff.com** (Figure 14).

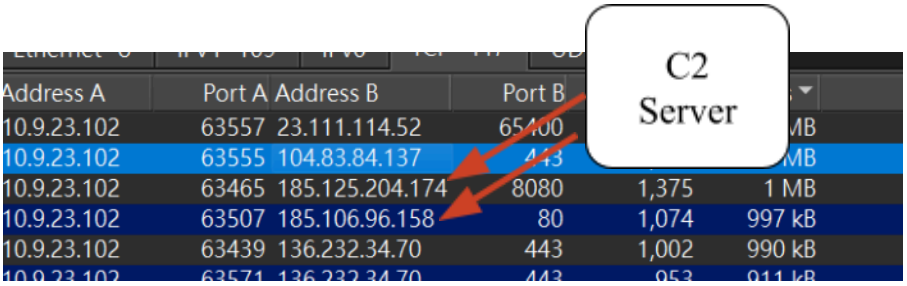


Figure 9. TCP stream statistics in order of packets

Figure 10. Virustotal review (1)

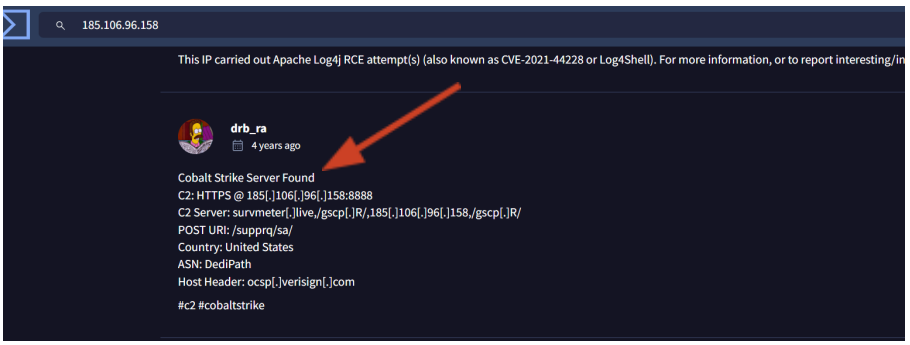


Figure 11. Virustotal review (2)

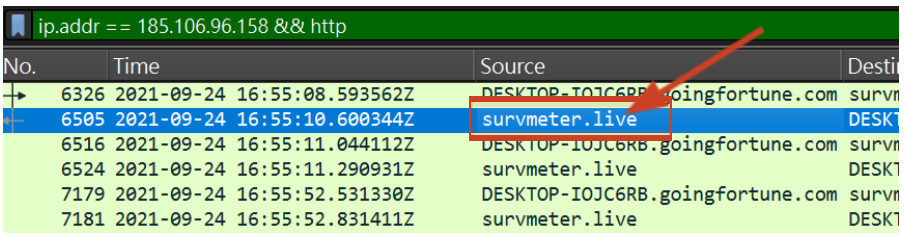


Figure 12. Domain linked to 185.106.96.158

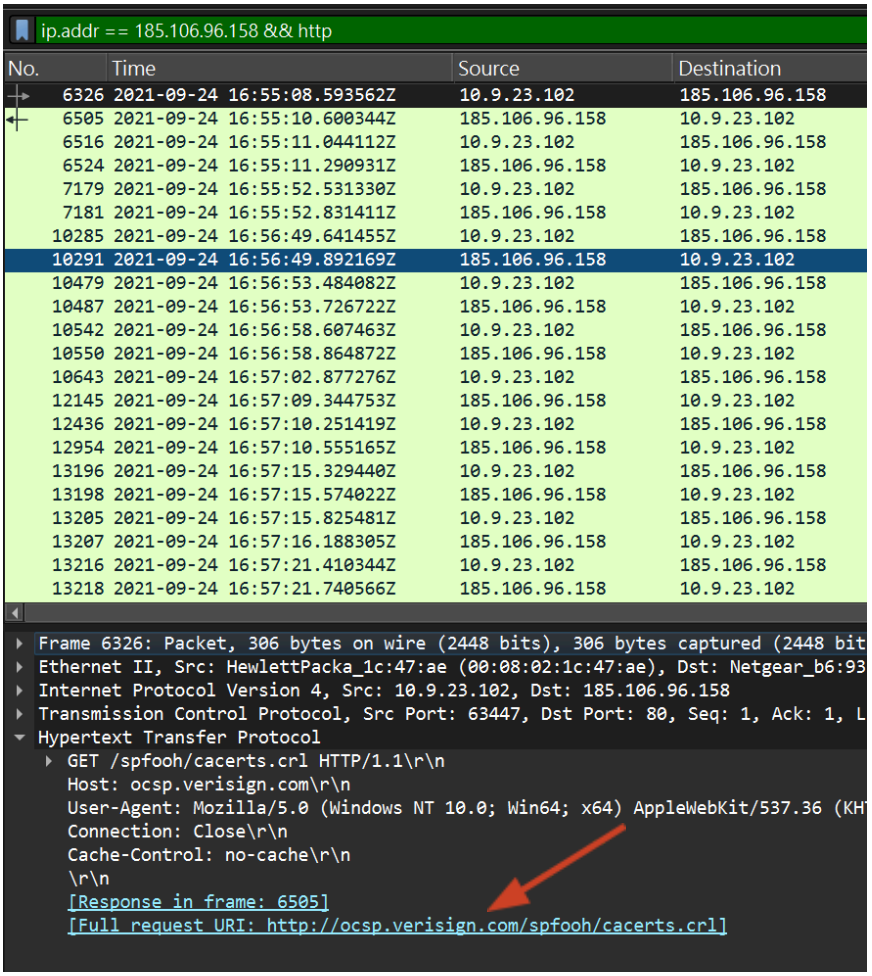


Figure 13. Request URL for 185.106.96.158

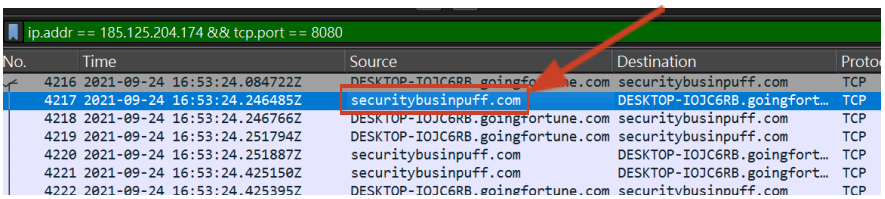


Figure 14. Domain linked to 185.125.204.174

Post-infection beaconing traffic was primarily directed to **maldivehost.net** (Figure 15), with the victim sending data beginning with **zLIisQRWZI9** (Figure 16). The first packet from the victim to the C2 server had a frame length of **281** bytes (Figure 16). Server-side response headers from **maldivehost.net** indicated Server: **Apache/2.4.49 (cPanel) OpenSSL/1.1.1l mod_bwlimited/1.4** (Figure 17), which further supports that this domain was attacker-controlled and actively responding to client beacons.

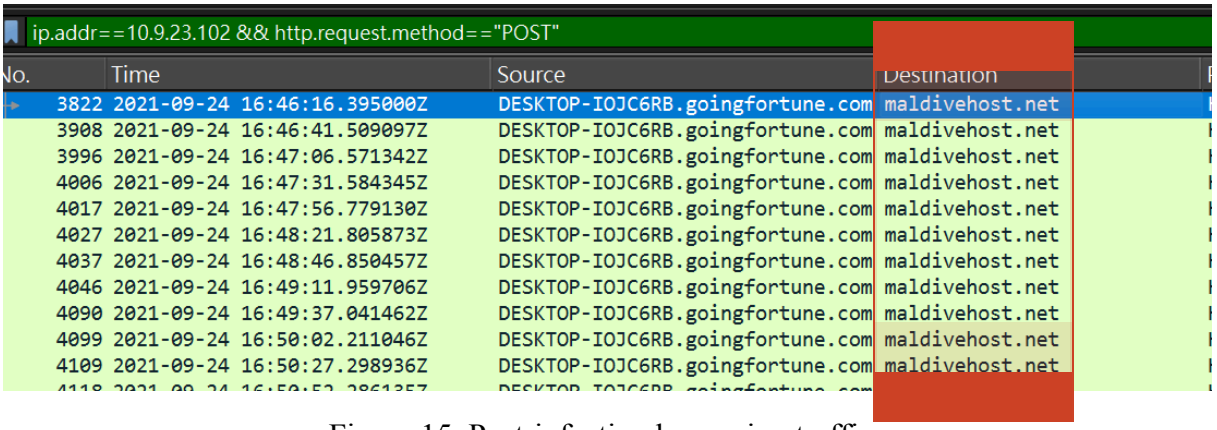


Figure 15. Post-infection beaconing traffic

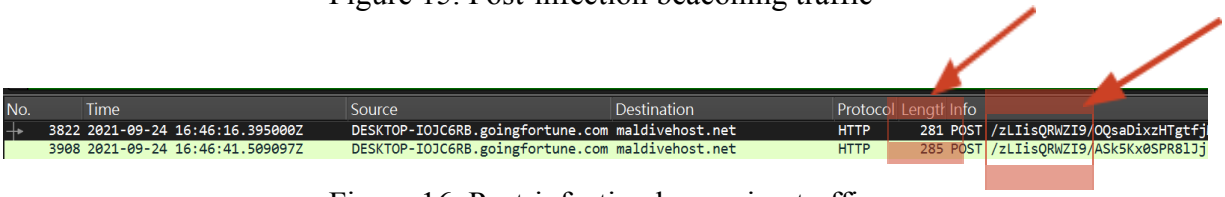


Figure 16. Post-infection beaconing traffic

Figure 17. Server header

Final exfiltration/check-in: The malware performed an external IP check via DNS at **2021-09-24 17:00:04**, querying **api.ipify.org** (Figure 18). SMTP traffic showed the first MAIL FROM address as **farshin@mailfa.com** (Figure 19). Following the same TCP stream, the password was transmitted using AUTH LOGIN (Base64), and the password associated with ho3ein.sharifi's password is **13691369** (Figures 20 & 21).

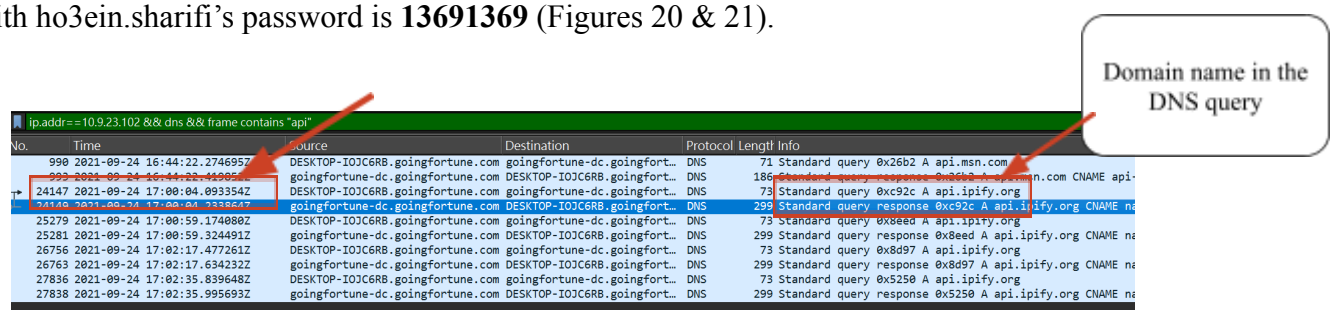


Figure 18. A DNS query occurred for the domain used by the malware

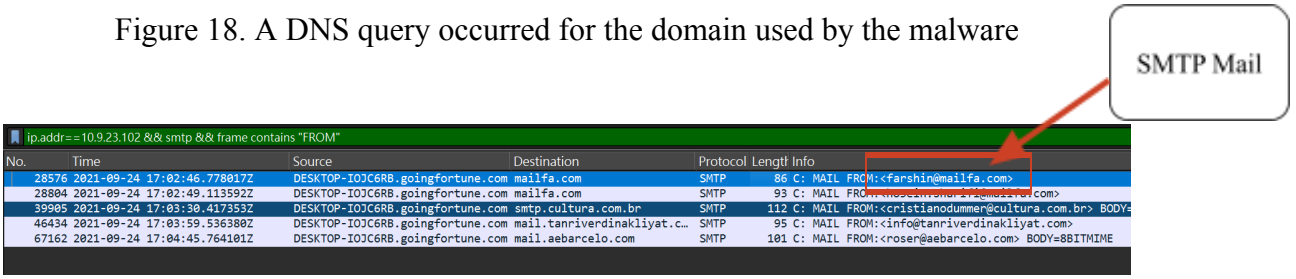


Figure 19. SMTP traffic

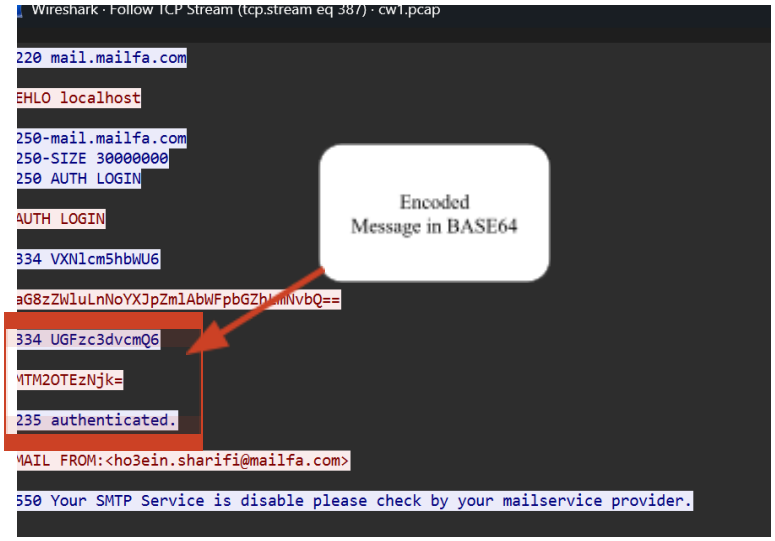


Figure 20. Packet content

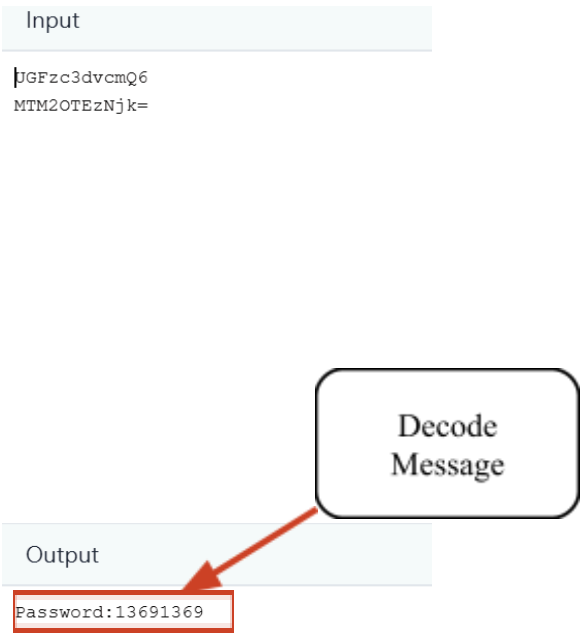


Figure 21. BASE64 decryption

Section 4 – Conclusion and References

4.1 Conclusion

This investigation identified the compromised host (10.9.23.102) and reconstructed the infection lifecycle observed in the PCAP. The incident began with the download of a malicious compressed archive containing an Excel file, which likely acted as the initial execution vector. The host subsequently established persistent command-and-control communication through HTTP POST beaconing and encrypted TLS connections, performed external IP discovery, and later initiated outbound SMTP authentication attempts, indicating that the system was repurposed for malspam or phishing activity.

To prevent the possibility of such an attack by stage:

Attack Stage	Evidence Observed	Targeted Prevention
Initial Access	HTTP download of ZIP	Web proxy filtering, file-type blocking, phishing protection, firewall rules, and antivirus scanning
Execution	XLS macro payload	Disable Office macros by default, user privilege restriction
C2	Obfuscated POST beacons	Egress filtering, TLS inspection, IDS/IPS monitoring
Recon	api.ipify.org lookup	DNS anomaly detection, threat intelligence feeds
Impact	SMTP spam	Outbound SMTP restrictions, email gateway monitoring

Table 2. Mapping of observed attack stages to preventative controls

Despite the implementation of these preventive measures, several open issues and challenges remain. A key challenge is the increasing prevalence of encrypted network traffic, which reduces payload visibility and complicates the detection of infection indicators. Another challenge lies in distinguishing legitimate but unusual user behaviour from malicious activity,

which may lead to false positives if not carefully contextualised. Additionally, modern malware may intentionally evade detection by mimicking normal traffic patterns or avoiding explicit disclosure of host identifiers.

In conclusion, while network-based intrusion analysis remains an effective approach for identifying compromised systems and reconstructing attack lifecycles, it must be complemented by layered defensive controls, secure configuration practices, and continuous monitoring to effectively prevent and respond to future incidents.

4.2 References

1. Palo Alto Networks Unit 42：
   Unit 42. (n.d.). Using Wireshark display filter expressions. Palo Alto Networks. https://unit42.paloaltonetworks.com/using-wireshark-display-filter-expressions/

2. Wireshark User's Guide：
   Wireshark Foundation. (2019). Wireshark user's guide (Version 4.7.0). https://www.wireshark.org/docs/wsug_html_chunked/

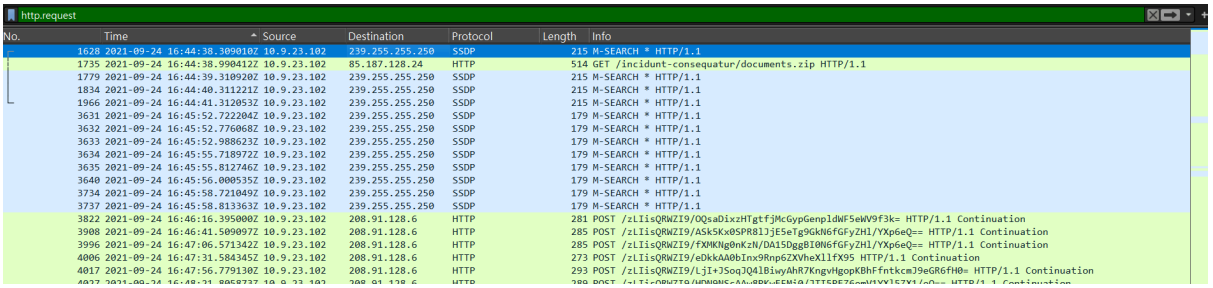
3. Brim Data. (2026). ZUI GitHub. https://github.com/brimdata/zui

Appendix

Part 1: Initial Infection & File Transfer

1. When did the initial malicious HTTP connection occur? (Provide the date and time in yyyy-mm-dd hh:mm:ss format).

Ans:



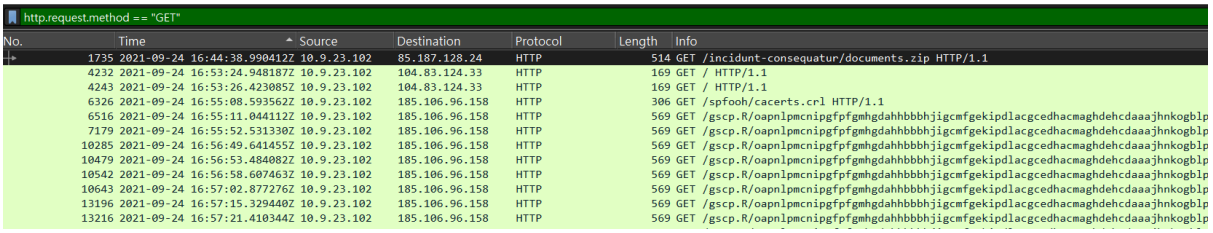
1735 2021-09-24 16:44:38.990412Z 10.9.23.102 85.187.128.24 HTTP 514 GET /incidunt-consequatur/documents.zip HTTP/1.1

UTC Arrival Time: Sep 24, 2021 16:44:38.309010000 UTC

2021-09-24 16:44:38

2. What is the name of the compressed file that the victim downloaded?

Ans:



[Full request URI: http://attirenepal.com/incidunt-consequatur/documents.zip]

documents.zip

3. Which domain hosted the malicious compressed file?

Ans:

attirenepal.com

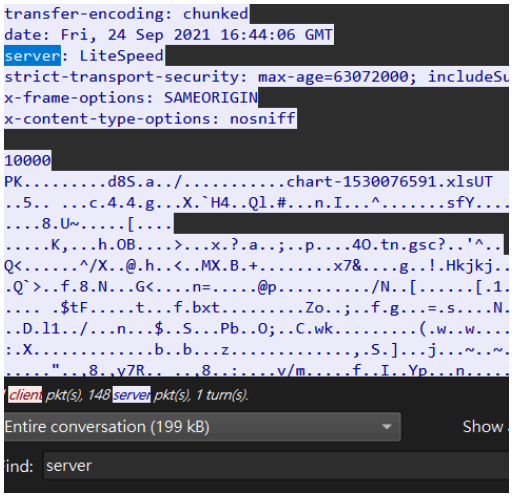
4. What is the name of the file located inside the compressed archive?

Ans:

chart-1530076591.xls

5. Identify the specific web server software (Server header) running on the malicious IP address that served the compressed file.

Ans:



LiteSpeed

6. What is the version number of the web server identified in the previous question?

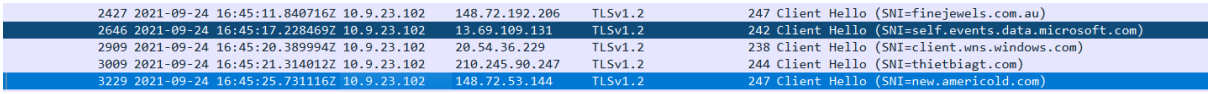
Ans:

PHP/7.2.34

7. Identify the three additional domains that were involved in downloading malicious files to the victim host.

Hint: Inspect HTTPS traffic and focus on the time window between 16:45:11 and 16:45:30. Note this range is in UTC, not BST.

Ans:



finejewels.com.au

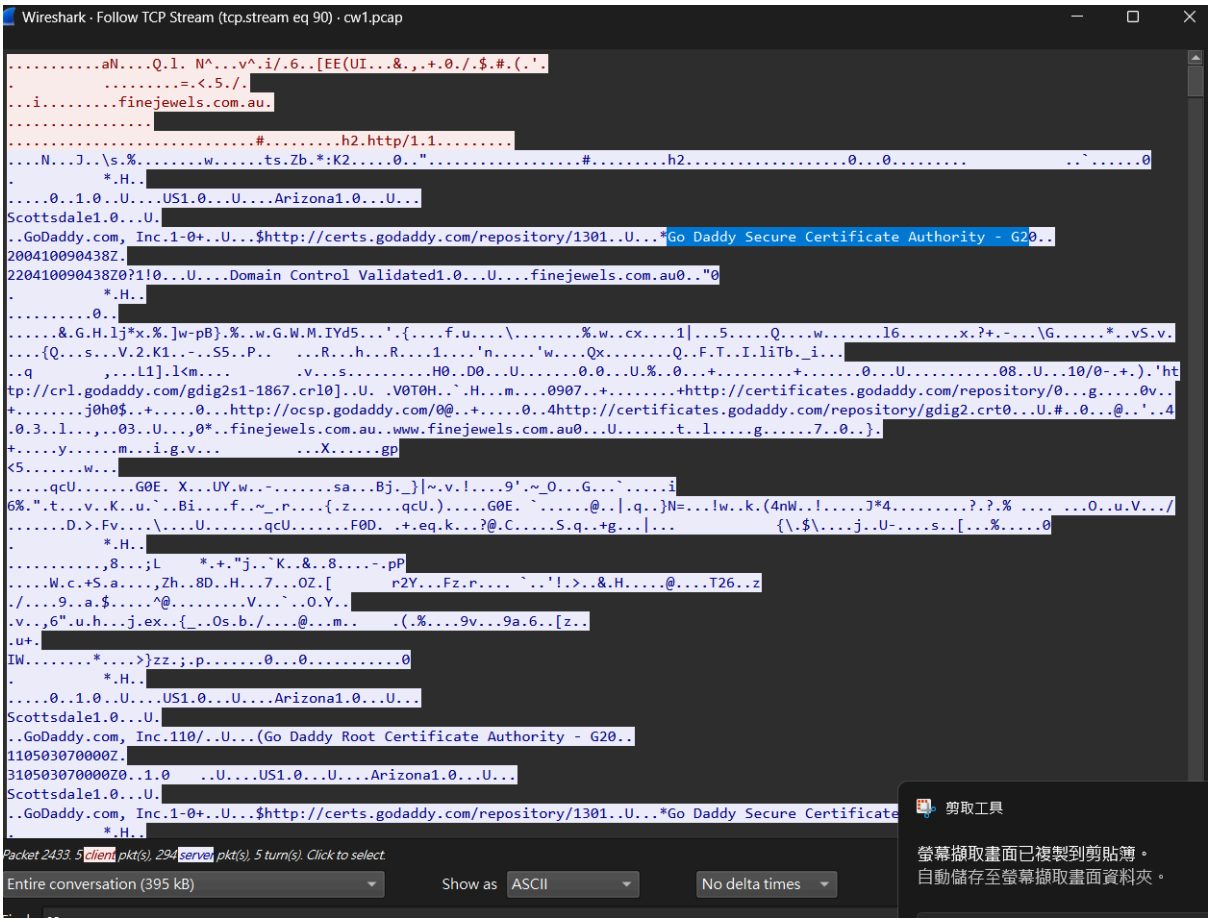
thietbiagt.com

new.americold.com

Part 2: Command and Control (C2) Activity

8. Which Certificate Authority (CA) issued the SSL certificate for the first domain identified in question 7?

Ans:



GoDaddy

9. What are the two IP addresses of the Cobalt Strike servers? (Provide them in sequential order).

Hint: Inspect the Conversations menu option

Ans:

Address A	Port A	Address B	Port B	Packets	Bytes
10.9.23.102	63557	23.111.114.52	65400	18,002	16 MB
10.9.23.102	63555	104.83.84.137	443	9,074	10 MB
10.9.23.102	63465	185.125.204.174	8080	1,375	1 MB
10.9.23.102	63507	185.106.96.158	80	1,074	997 kB
10.9.23.102	63439	136.232.34.70	443	1,002	990 kB

Virustotal

185.106.96.158

185.125.204.174

10.What is the value of the Host header for the first Cobalt Strike IP address?
Hint: Apply a filter to isolate DNS queries.

Ans:
ocsp.verisign.com

11.What is the domain name associated with the first Cobalt Strike IP address?
Hint: Take a closer look at HTTPS (443)

Ans:
survmeter.live

12.What is the domain name associated with the second Cobalt Strike IP address?
Hint: Apply a filter to capture HTTP POST requests.

Ans:
securitybusinpuff.com

13.What is the domain name used for the post-infection traffic?

Ans:
maldivehost.net

14.What are the first eleven characters of the data the victim host sends to the malicious domain identified in the previous question?

Ans:
zLIisQRWZI9

15.What was the length of the first packet the victim sent to the C2 server?

Ans:
281

16.What was the Server header value for the malicious domain from question 13?

Ans:

Apache/2.4.49 (cPanel) OpenSSL/1.1.1l mod_bwlimited/1.4

Part 3: Final Exfiltration/Check-in

17.What was the date and time (in yyyy-mm-dd hh:mm:ss UTC format) when the DNS query occurred for the domain used by the malware to check the victim's external IP address?

Ans:

2021-09-24 17:00:04

18.What was the domain name in the DNS query from the previous question?

Ans:

api.ipify.org

19.What was the first email address observed in the SMTP traffic in the pcap file (the MAIL FROM address)?

Ans:

28576 2021-09-24 17:02:46.778017Z 10.9.23.102 185.4.29.135 SMTP 86 C: MAIL FROM:<farshin@mailfa.com>

farshin@mailfa.com

20.Follow the stream from Q19. What is ho3ein.sharifi’s password?

Ans:

220 mail.mailfa.com

EHLO localhost

250-mail.mailfa.com
250-SIZE 30000000
250 AUTH LOGIN

AUTH LOGIN

334 VXNlcm5hbWU6

ZmFyc2hpbkBtYWlsZmEuY29t

334 UGFzc3dvcmQ6

ZGluYW1pdA==

235 authenticated.

MAIL FROM:<farshin@mailfa.com>

550 Your SMTP Service is disable please check by your mailservice provider.

Due to the communication using BASE64, which is decryptable, the password was found in the decrypt output.

Password:**13691369**

Run the query below to get the clear details of the alert event:

```
event_type==”alert” | cut
ts,src_ip,dest_ip,dest_port,flow_id,alert.signature,alert.severity
,alert.category | sort -r ts
```

The result can be found on the GitHub repository (https://github.com/aka331/COMP3010HK/blob/main/CW1/event_alert.csv)

Timeline (key events with exact times)

All times are from the PCAP timestamps.

1. **Initial download (infection vector)**

- **16:44:38.990** — Victim downloads a ZIP over HTTP:
  GET /incidunt-consequatur/documents.zip from **attirenepal.com** (IP: **85.187.128.24**)

Extracted that file from the PCAP, and it contains:

- chart-1530076591.xls (inside documents.zip)

This is a classic pattern: **ZIP → Excel lure → macro/dropper execution**.

2. **First C2 / malware check-in**

- **16:46:16.395** — Victim starts POST beacons to **maldivehost.net** (IP: **208.91.128.6**) with highly obfuscated URI paths like:
  POST /zLIisQRWZI9/<base64-ish blob>

This looks like **malware C2 polling/tasking**.

3. **Second C2 channel over TLS**

- **16:53:28.188** — Victim initiates TLS sessions to **185.125.204.174:8080** with SNI: **securitybusinpuff.com**
  Certificate chain shows **Let’s Encrypt (R3)**, which is common for attacker infrastructure too.

4. **External IP discovery**

- **17:00:04.093** — DNS query for **api.ipify.org**
  This is often used by malware to learn the victim's **public IP**.

5. **Botnet spamming/phishing begins**

- **17:02:43.150** — Clear SMTP commands observed from victim to external mail servers (port **25**), e.g., EHLO localhost
- The victim then performs many SMTP deliveries to different domains/servers (direct-to-MX behavior).

The victim host (10.9.23.102) became infected and began communicating with a known malware loader infrastructure. ZUI/Suricata generated repeated "ET MALWARE SQUIRRELWAFFLE Loader Activity (POST)" alerts for HTTP POST beacons to 208.91.128.6 starting at 16:46Z, followed by "SQUIRRELWAFFLE Server Response" alerts indicating successful C2 responses. Shortly after, the host initiated multiple TLS sessions that matched an "ET JA3 Hash – Possible Dridex" fingerprint (including traffic to 185.125.204.174:8080 and 136.232.34.70:443), suggesting a second-stage bot/trojan component. The host then performed external IP discovery via api.ipify.org. Finally, SMTP anomalies and email-delivery behavior indicate the compromised machine was used to send malspam/phishing emails (often via direct SMTP to external servers), consistent with botnet/spambot activity.

Student Declaration of AI Tool use in this Assessment

Please indicate your level of usage of generative AI for this assessment - please tick the appropriate category(s).

If the "Assisted Work" or "Partnered Work" category is selected, please expand on the usage and in which elements of the assignment the usage refers to.

Solo Work	**S1 - Generative AI tools have not been used for this assessment.**	☐
Assisted Work	**A1 – Idea Generation and Problem Exploration** Used to generate project ideas, explore different approaches to solving a problem, or suggest features for software or systems. Students must critically assess AI-generated suggestions and ensure their own intellectual contributions are central.	☐
	A2 - Planning & Structuring Projects AI may help outline the structure of reports, documentation and projects. The final structure and implementation must be the student's own work.	☐
	A3 – Code Architecture AI tools maybe used to help outline code architecture (e.g. suggesting class hierarchies or module breakdowns). The final code structure must be the student's own work.	☐
	A4 – Research Assistance Used to locate and summarise relevant articles, academic papers, technical documentation, or online resources (e.g. Stack Overflow, GitHub discussions. The interpretation and	☐

	integration of research into the assignment remain the student's responsibility.	
	A5 - Language Refinement Used to check grammar, refine language, improve sentence structure in documentation not code. AI should be used only to provide suggestions for improvement. Students must ensure that the documentation accurately reflects the code and is technically correct.	☐
	A6 – Code Review AI tools can be used to check comments within the code and to suggest improvements to code readability, structure or syntax. AI should be used only to provide suggestions for improvement. Students must ensure that the code accurately reflects their knowledge and is technically correct.	☐
	A7 - Code Generation for Learning Purposes Used to generate example code snippets to understand syntax, explore alternative implementations, or learn new programming paradigms. Students must not submit AI-generated code as their own and must be able to explain how it works.	☐
	A8 - Technical Guidance & Debugging Support AI tools can be used to explain algorithms, programming concepts, or debugging strategies. Students may also help interpret error messages or suggest possible fixes. However, students must write, test, and debug their own code independently and understand all solutions submitted.	☐
	A9 - Testing and Validation Support AI may assist in generating test cases, validating outputs, or suggesting edge cases for software testing. Students are	☐

	responsible for designing comprehensive test plans and interpreting test results.	
	A10 - Data Analysis and Visualization Guidance AI tools can help suggest ways to analyse datasets or visualize results (e.g. recommending chart types or statistical methods). Students must perform the analysis themselves and understand the implications of the results.	☐
	A11 - Other uses not listed above Please specify:	☐
Partnered Work	**P1 - Generative AI tool usage has been used integrally for this assessment** Students can adopt approaches that are compliant with instructions in the assessment brief. Please Specify: - The analysis of the pcap file to verify the answer of founding - README.md - report drafting and improvement - generate suggested expression in Wireshark	☒

Please provide details of AI usage and which elements of the coursework this relates to:
improve the report, pcap analysis

I understand that the ownership and responsibility for the academic integrity of this submitted assessment falls with me, the student.	☒
I confirm that all details provide above are an accurate description of how AI was used for this assessment.	☒