Coursework 1: Set Exercises - PCAP Analysis

Github Link: <https://github.com/Mattrfish/COMP3010-SecOps-and-Incident-Management.git>

Youtube Walkthrough Link:

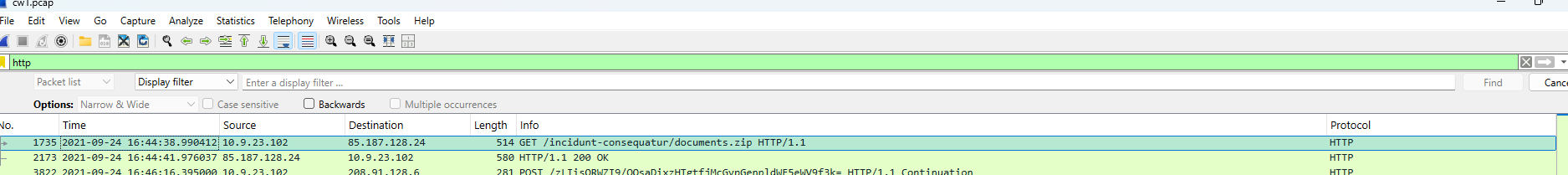
# Introduction

This incident report presents a detailed analysis of a provided PCAP file to investigate a suspected intrusion on the network. The purpose of this analysis is to identify the infected system, determine how the infection occurred, describe what type of attack was involved, and highlight the key indications of compromise (IOCs).

This report is structured as follows: The methodology section outlines what tools and techniques that will be used throughout the analysis and explains how they were applied to investigate the intrusion. The results section presents the key findings that I have gathered from the PCAP analysis, supported by evidence and screenshots. Finally, the conclusion summarises the overall findings and discusses prevention strategies to mitigate similar incidents in the future, and it reflects on any issues or challenges that were encountered during the investigation.

# Methodology

For this intrusion analysis I am going to be using Wireshark to analyse the pcap file as it is the most effective open source tool for analysing network traffic at the packet-level. The goal of the investigation was to determine which host was infected, how the infection occurred, and the type of malware or attack involved.



A screenshot of a computer program

AI-generated content may be incorrect. A screenshot of a computer

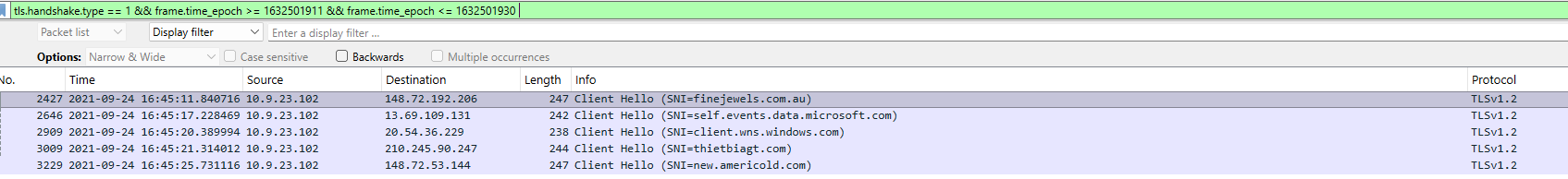
AI-generated content may be incorrect.

The analysis began by inspecting HTTP traffic using the “http” display filter, which helped reveal suspicious activity like potential malicious file downloads. I used the “follow http stream” feature, to view complete communications between the internal host and external domains, allowing me to trace the infection process and inspect connections further by examining server information, headers, and communication patters. It also enabled me to inspect transferred file types without exporting them, reducing the risk of executing malicious content on my local machine.

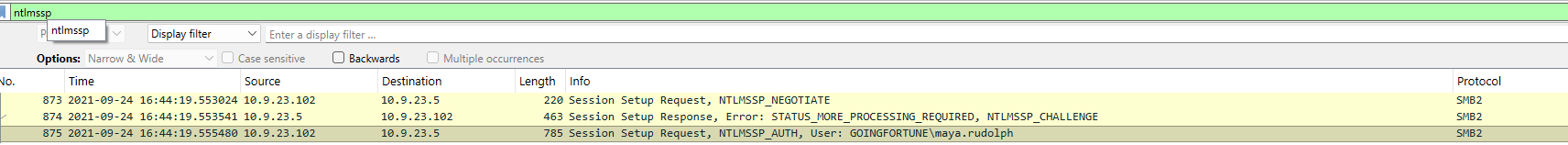
A screenshot of a computer

AI-generated content may be incorrect.

The packet pane provided further insight into hostnames, server information, and response headers. Individual packets were followed in this pane to the Ethernet 2 layer, revealing the source and destination MAC address at the link layer. This provided additional host identification.



To inspect specific elements of the timeline and analyse key stages of the infection, the “frame.time\_epoch” filter was applied, along with the “tls” filter through the https traffic.



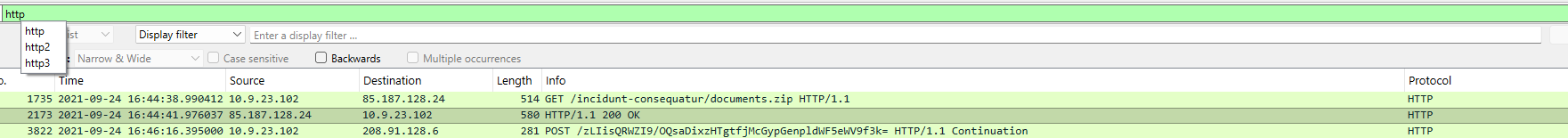
A computer screen shot of a computer code

AI-generated content may be incorrect.

The “ntlmssp” and “smb” filters were applied to locate authentication packets, which revealed the infected devices hostname and user account name through NTLM logins on a domain.

These features enabled me to reconstruct the intrusion timeline, identify the compromised host and understand the method and type of infection that occurred.

# Results



A screenshot of a computer program

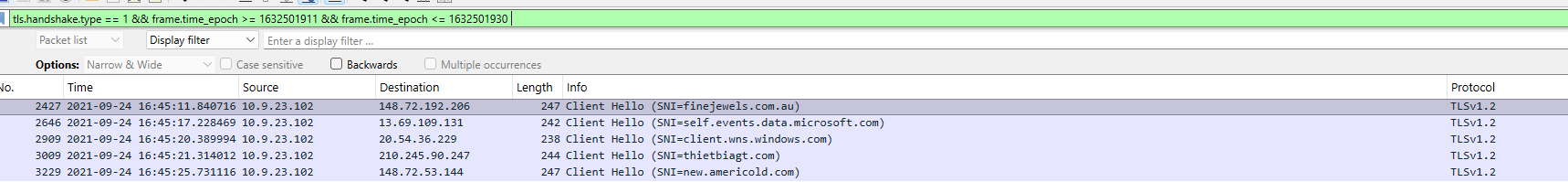
AI-generated content may be incorrect.

The filtered the data to find the initial http connection which occurred on 2021-09-24 16:44:38 between the victim host (10.9.23.102) and destination server (85.187.128.24). Following the http stream, it was discovered that the victim downloaded a compressed file named documents.zip from the domain attirenepal.com, which contained a file called chart-1530076591.xls. The destination server was identified as running LiteSpeed web server software with version PHP /7.2.34.

I decided to go back and inspect to http timeline further, which shows a POST request at 16:46:16. I decided to look closer into the minute between the initial HTTP OK response and the POST request to investigate the suspicious activity.

A screenshot of a computer

AI-generated content may be incorrect.



Straight away I could see there were some suspicious looking TLS requests so I filtered these out to ClientHello packets only and revealed three additional domains that were involved in downloading malicious files to the victim host, which were finejewels.com.au, thietbiagt.com and new.americold.com.

# Conclusion

# References

# Appendix