**SPR600**

**Final Individual Projec**t

Incident Report

Name: Ananthu Krishna Vadakkeppatt

Student ID: 154290217

Tools Used:

1. ELK
2. Wireshark
3. Zeek
4. Suricata
5. Pcapmonkey

Table of Contents

[Executive Summary 2](#_Toc1115824579)

[Detailed Findings 3](#_Toc1862858521)

[Indicators of Compromise (IOC) 8](#_Toc422731369)

[Threat Hunting 9](#_Toc1540697024)

[Methodology 9](#_Toc467502102)

[Implementation 9](#_Toc1221812269)

[Considerations and Limitations 14](#_Toc1400914203)

[References 14](#_Toc1191526862)

# **Executive Summary**

On Saturday the 6th of April 2024, at around 3:13 PM the Optus company’s network activity was seen to contain a lot of requests. This was the start of an attack that was successful in exfiltrating numerous customer credentials through the help of an open API that was administrated without any user authentication measures. The IP of the infected machine was 192.168.10.10 and the hostname was APIMachine. The user identified from this machine was named Optus. Moreover, the cause of this infection wasn’t malware, as this was a result of a human error that led to the attacker exploiting a configuration vulnerability. The infected machine's operating system was seen as of a Linux distribution that was Ubuntu. The breach type of this security incident is configuration vulnerability, and the severity of this attack is High.

# **Detailed Findings**

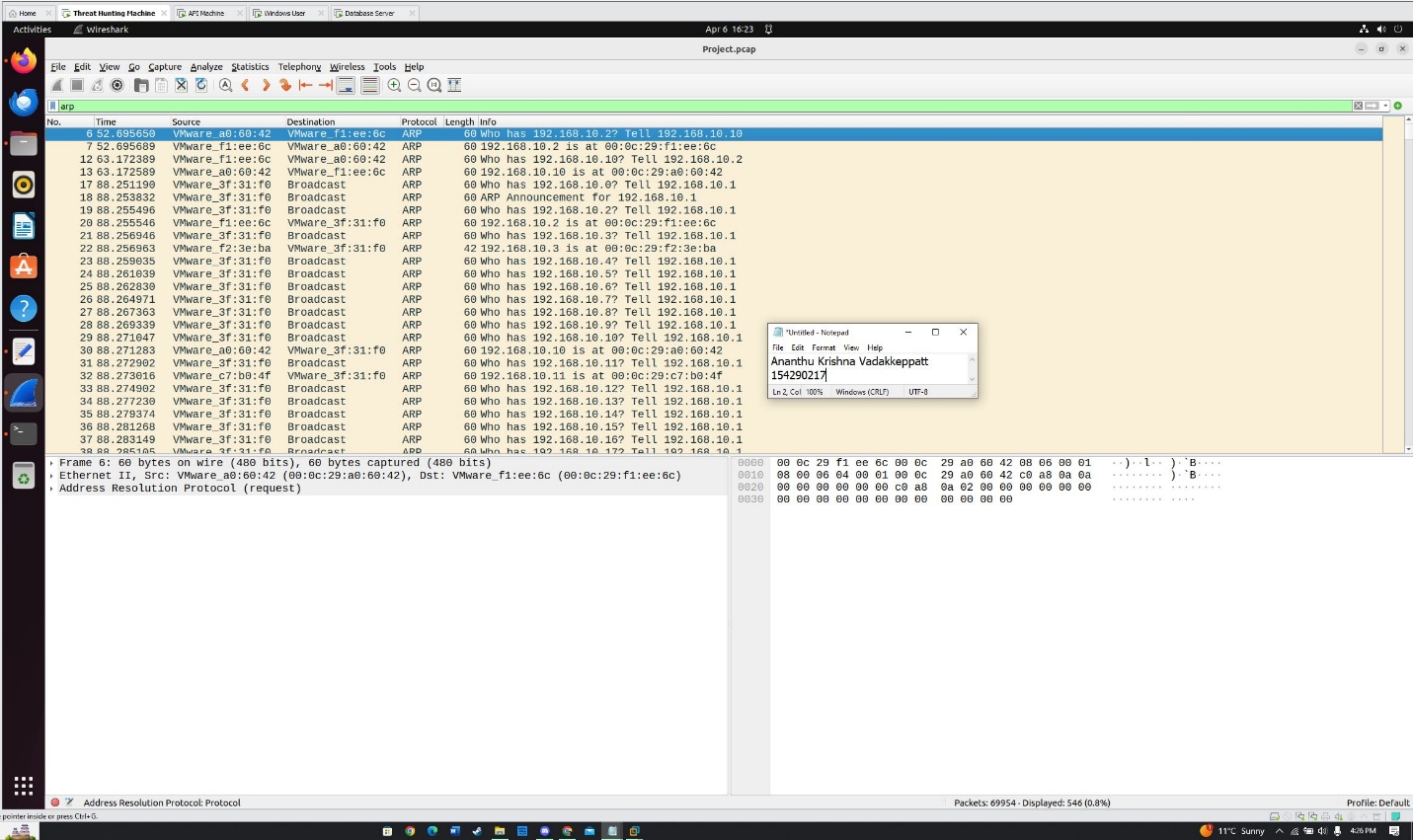
While analyzing the network capture of this security incident, I discovered various methods used by the attacker to successfully carry out this attack. The start of the packet capture can be seen with many ARP requests being sent from IP 192.168.10.1 to the network of 192.168.10.0/24. Hence, it is my assumption that the attacker at this phase is carrying out an Arp scan within the network. The main goal of the Arp scan remains to discover active hosts within the network. Arp requests can be seen being sent to network 192.168.10.0/24 as evidence to support this claim.

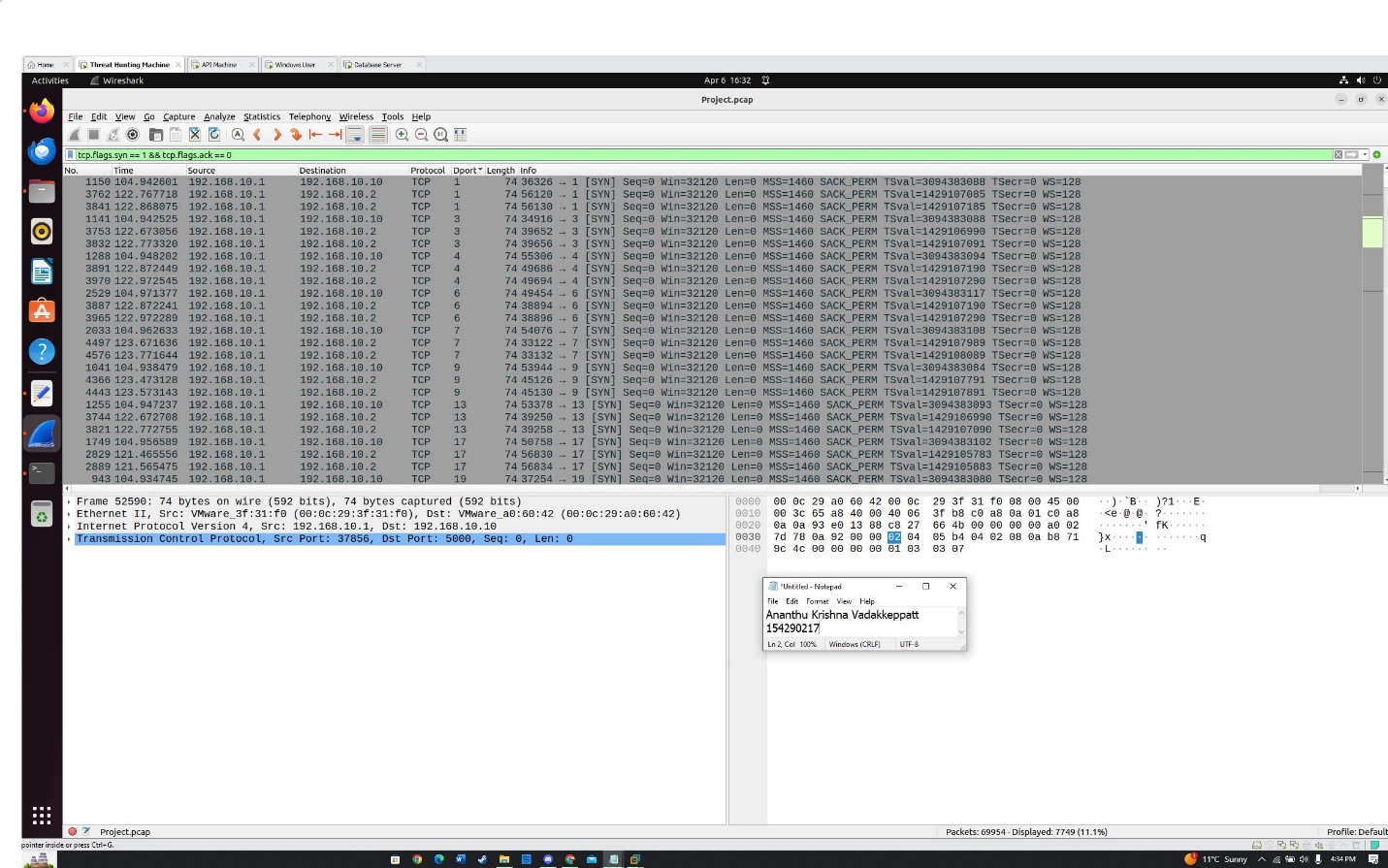
Once the attacker had the IP addresses of the hosts active within the network, the packet capture suggests that the attacker resorted to scanning the IP addresses with the Nmap tool. Evidence of the same can be seen within the capture, where the IP 192.168.10.1 is only focusing on sending probes to ports of the IP addresses without checking if they were active. There are several DNS records seen after this section within the packet capture, this suggests that the attacker might have tried to communicate or attain further information related to the domain which is seen to be optus.ca. A little later, I was able to see a host of HTTP GET requests made to port 80 as well port 5000 of the machine with the IP 192.168.10.10.

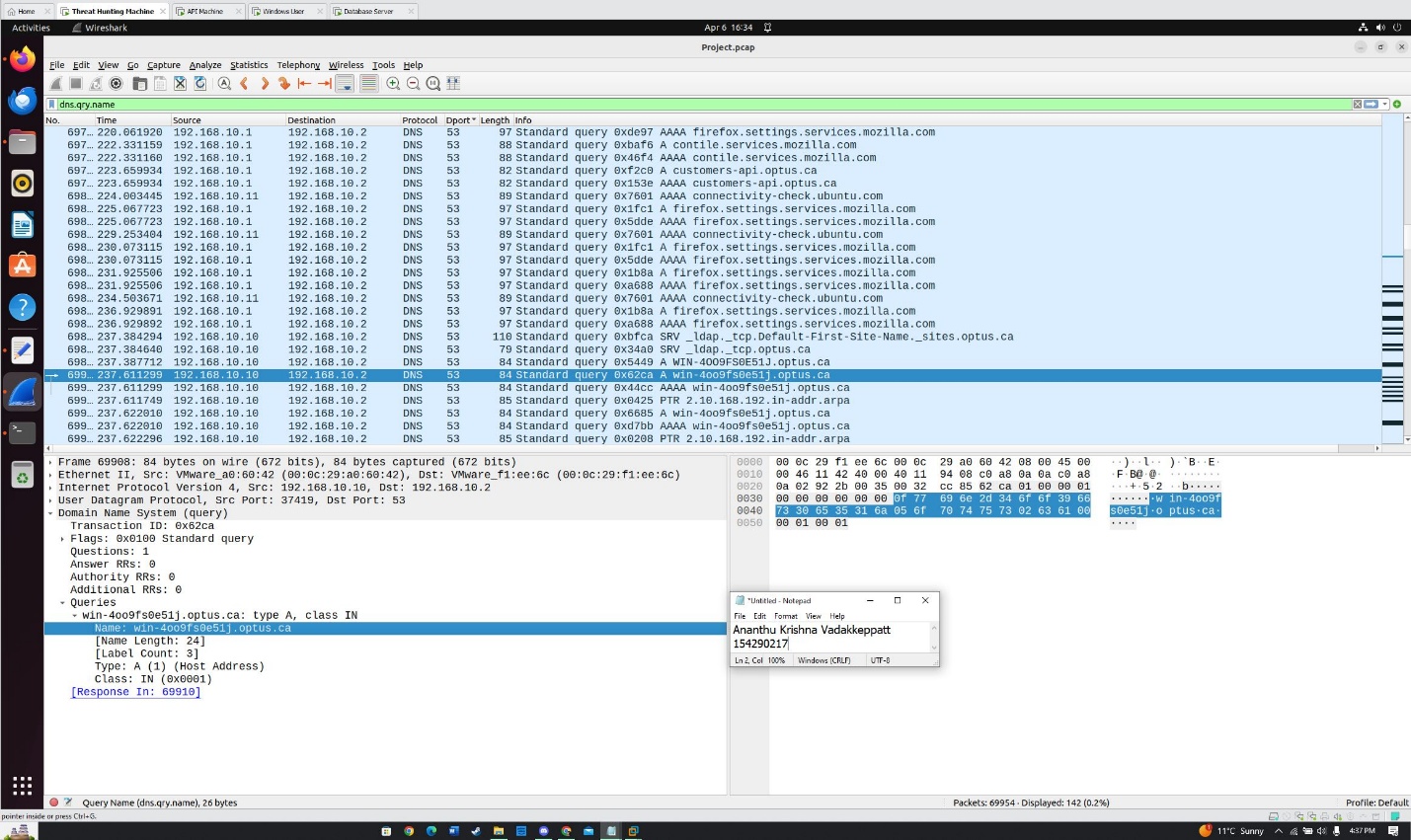
On further examinations of the requests, the information tab showed that the attacker was trying to access different endpoints or pages that could be found within both those web applications. Most of these requests did return a 404 Not Found error. Hence, I was able to conclude that the attacker was launching a directory traversal attack on these web applications that were seen to be open. If the attacker was to find something one of the requests would very obviously have returned a 200 OK response code.

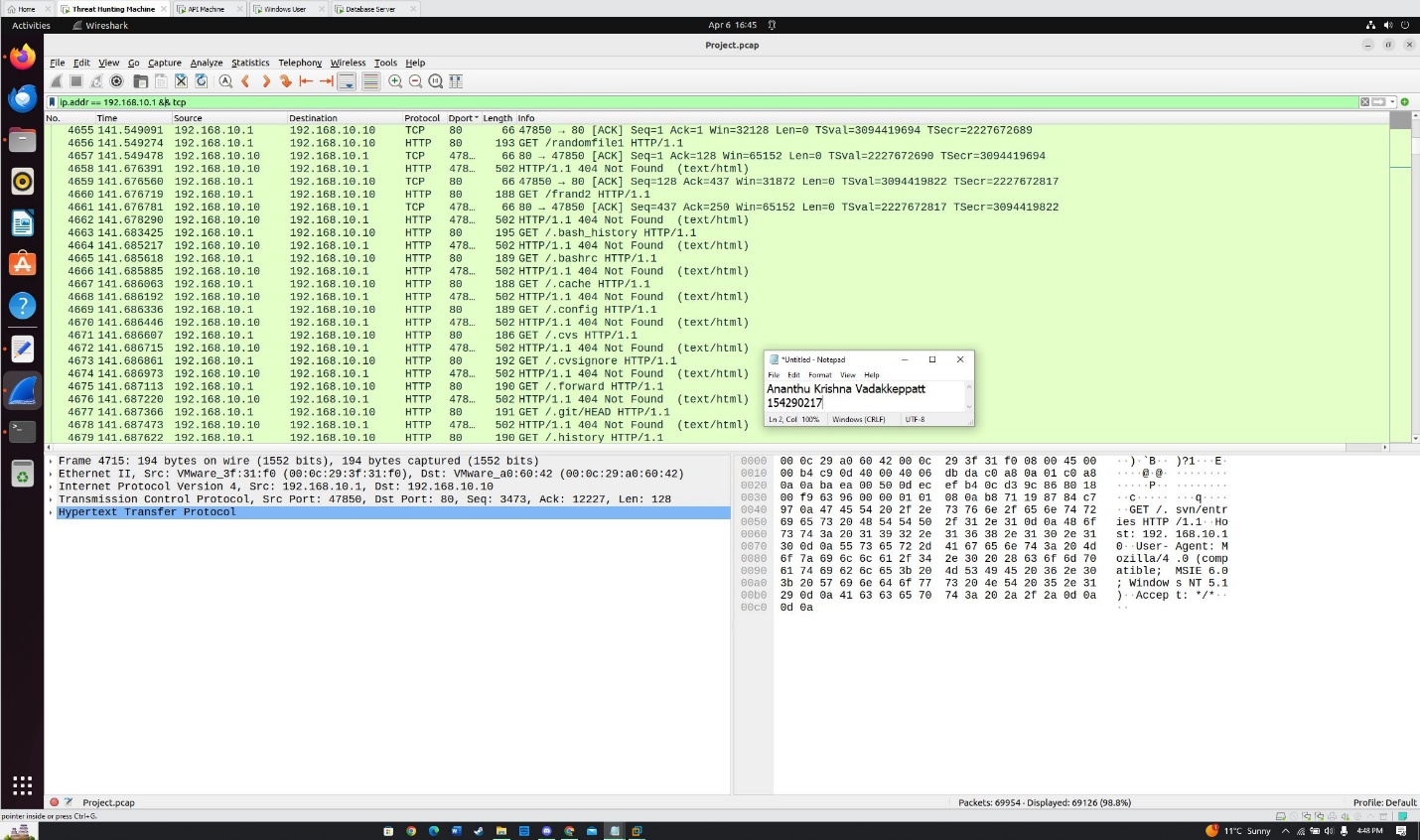
Therefore, I filtered the connections using the attacker’s IP as well the TCP protocol that is used by the tool. I was able to find one request related to the web application running on port 5000 that returned a 200 OK response. The endpoint was /customers, and with further analysis of the packet, I could see the contents of the same. Customer records that included columns such as Name, Home Address, Phone Number, Date of Birth and other sensitive information including Drivers License numbers were seen to be retrieved by the attacker. It was clear that the attacker’s next step in this attack would be to look at methods through which aid in the exfiltration of the data that was attained.

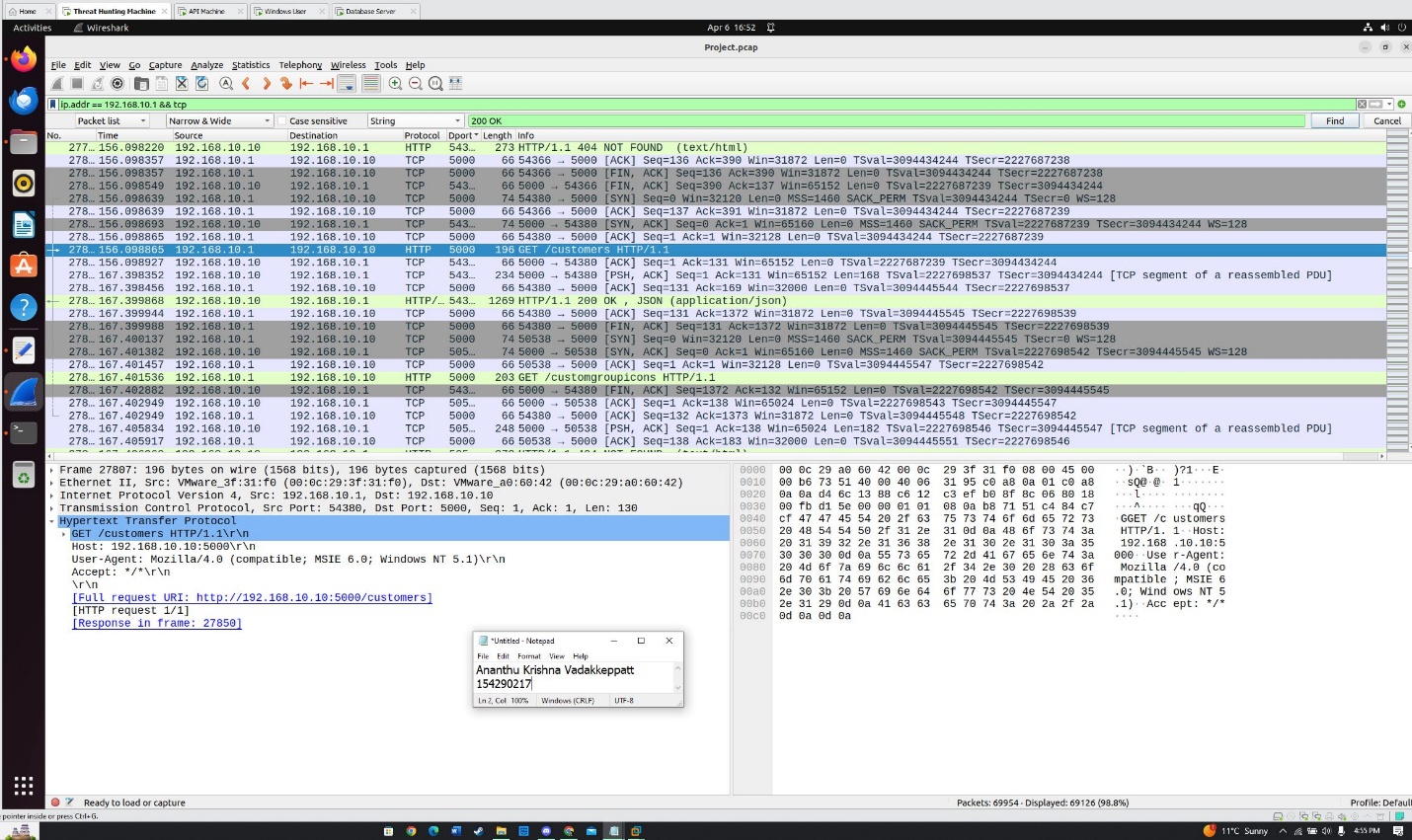
As there seems to be no authentication mechanisms set up on this web application and the database is seen not to employ unique Customer IDs, the attacker is suspected of running a script that has helped him to exfiltrate the information to his machine.

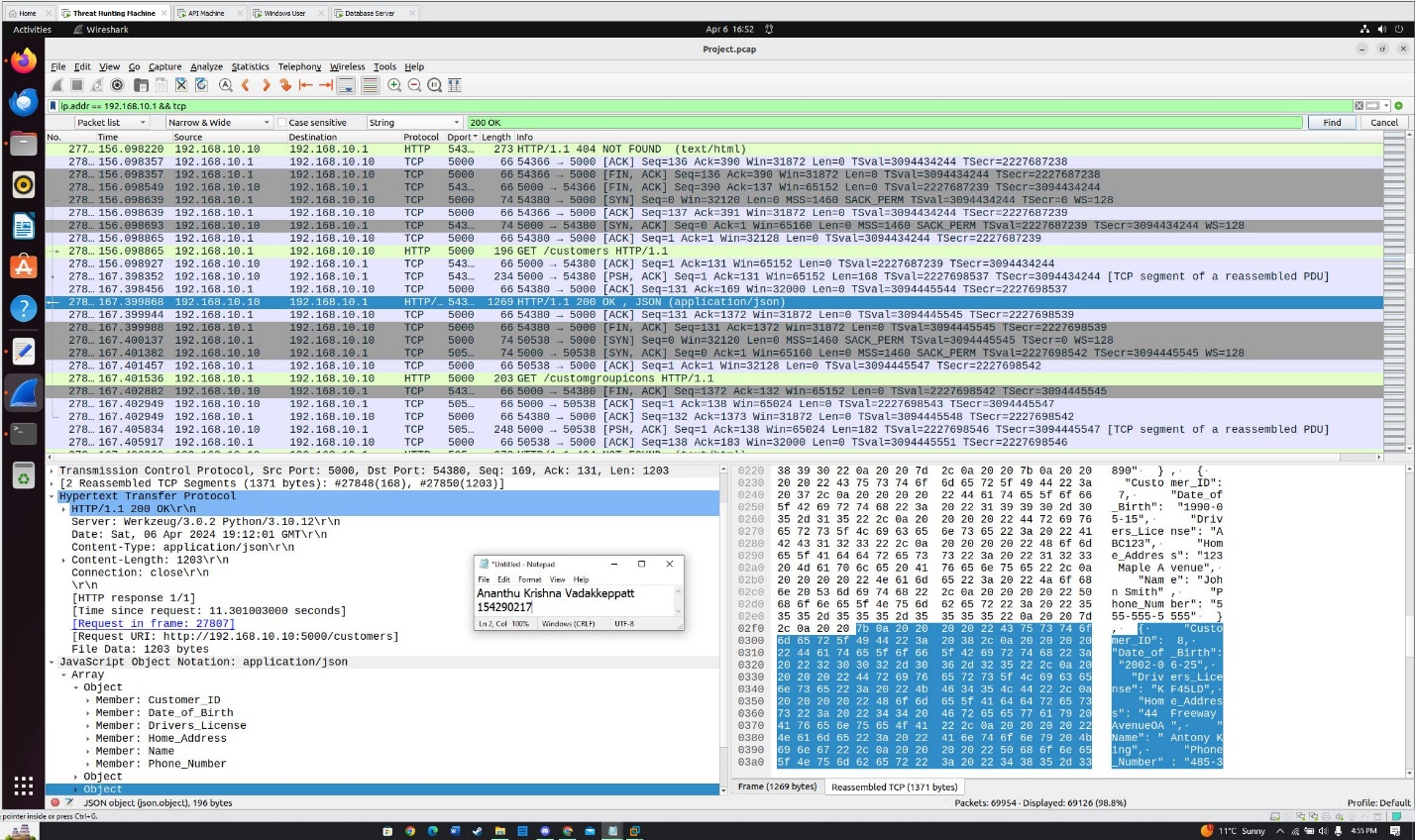
Fig.1 Arp Scan performed by the attacker.

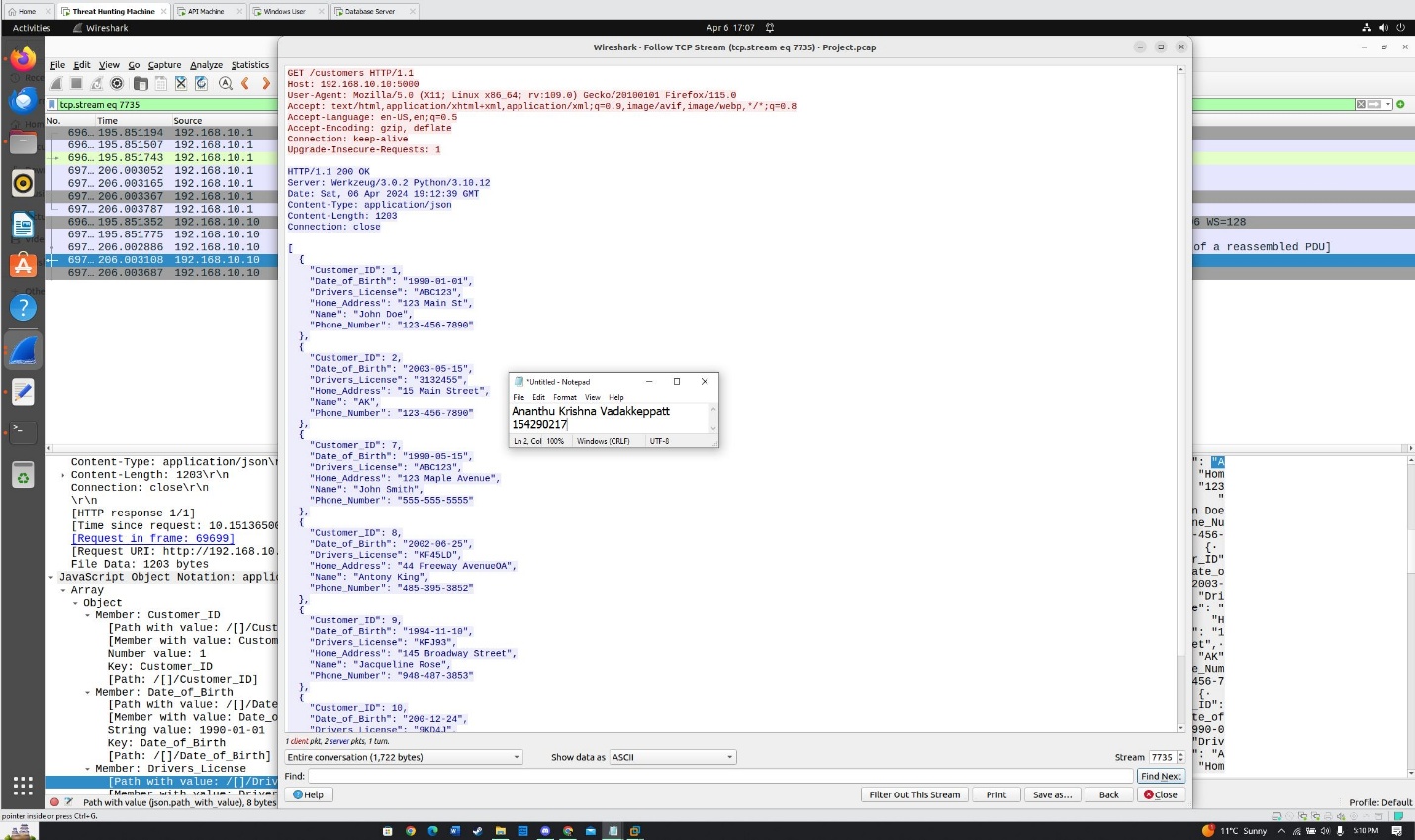
Fig.2 Traces of Nmap Scans on the IP Addresses.

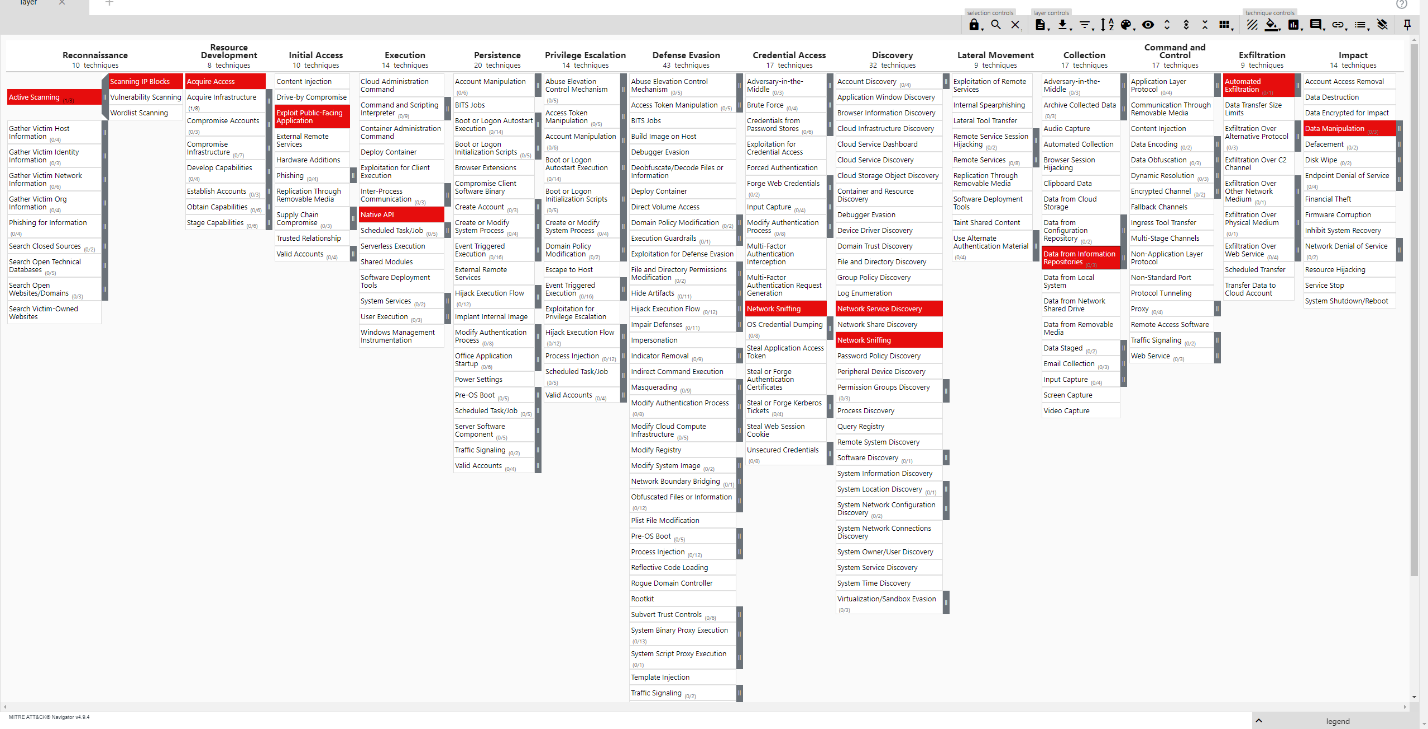
Fig.3 DNS query made by the attacker to the Domain.

Fig.4 Directory Traversal attempts by the attacker on the web applications.

Fig.5 The Customers endpoint being identified by the directory traversal tool.

Fig.6 The web application is seen to be retrieved and exposes the credentials.

Fig.7 User Credentials extracted by the attacker.

Fig. 8 Mitre Attack Framework for the Security Incident.

# **Indicators of Compromise (IOC)**

**Attacker’s IP:** 192.168.10.1

**Attacker MAC Address:** 00:0c:29:3f:31:f0

**Victims' IP:** 192.168.10.10

**Victim’s MAC Address:** 00:0c:29:a0:60:42

**API Exposed:** http://192.168.10.10/customers

**Host Used by the Victim:** Ubuntu

**Broswer Used by The Attacker:** Mozilla/4.0

**Start Time and Date of the Attack:** 15:13, 6th Saturday 2024

# **Threat Hunting**

## Methodology

Our main goal in this incident report was to go over the security incident and analyze the security vulnerabilities and attacking methods that were employed by the attacker. Thus, the first and foremost step that I took was to investigate the network activity that was available with the network capture using Wireshark. Due to the absence of any malware, threat actors and other vulnerabilities in this security incident, network activity was the focus of my investigation. The network activity exposed many techniques used by the attackers to get to the open API and hence exfiltrate customer-sensitive information. Moreover, I used the ELK stack to support my claims of how the attacker led this attack. Furthermore, the intel attained from the Zeek logs helped the threat hunting process in identifying various Indicators of Compromise as well as authenticity to other sources such as Wireshark and ELK. These kinds of threat hunts can be automated with the use of various scripts or tools that can be tuned to understand patterns which are associated with each type of attack. The problem as well as the results can be well presented if the report is direct and includes the right mix of technical and factual knowledge.

## Implementation

Throughout this phase of incident investigation, I employed various threat hunting techniques that were useful to achieve the end goal of identifying the intent of the attacker. We started by looking through Wireshark network activity that was captured. Kibana that was deployed using the Pcapmonkey gave me options to go through the network activity and slide through the alerts that were triggered along with the general overview of the entire attack. Moreover, Kibana was useful in creating visualizations and signatures that would allow us to identify and trace the steps of the attack if it ever repeats again. Suricata and Zeek services were used to create signatures to detect stages of the attack not identified by the default rules among both tools. Rules created by both tools are then

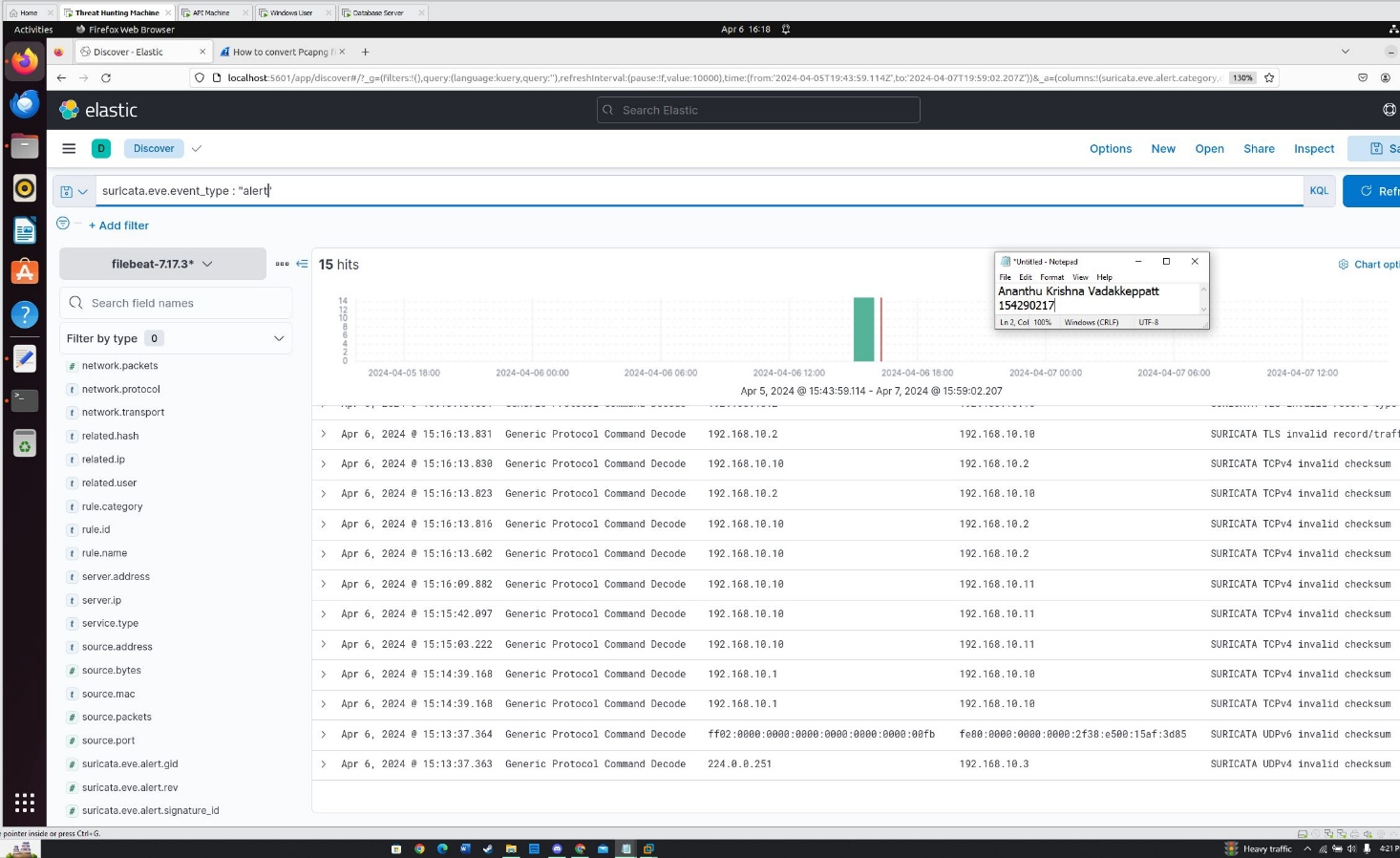


Fig.9 Using ELK to analyze the network activity of the Pcap.

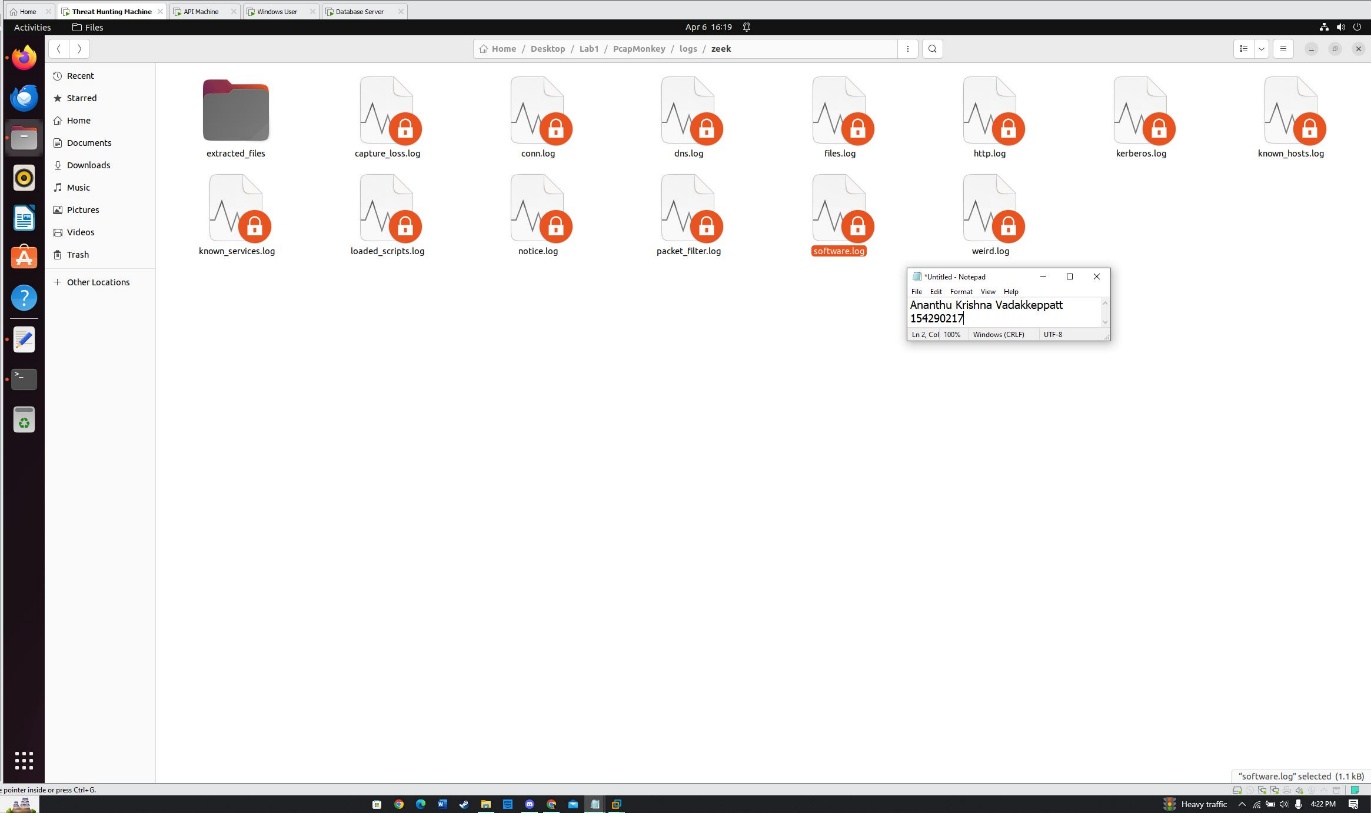
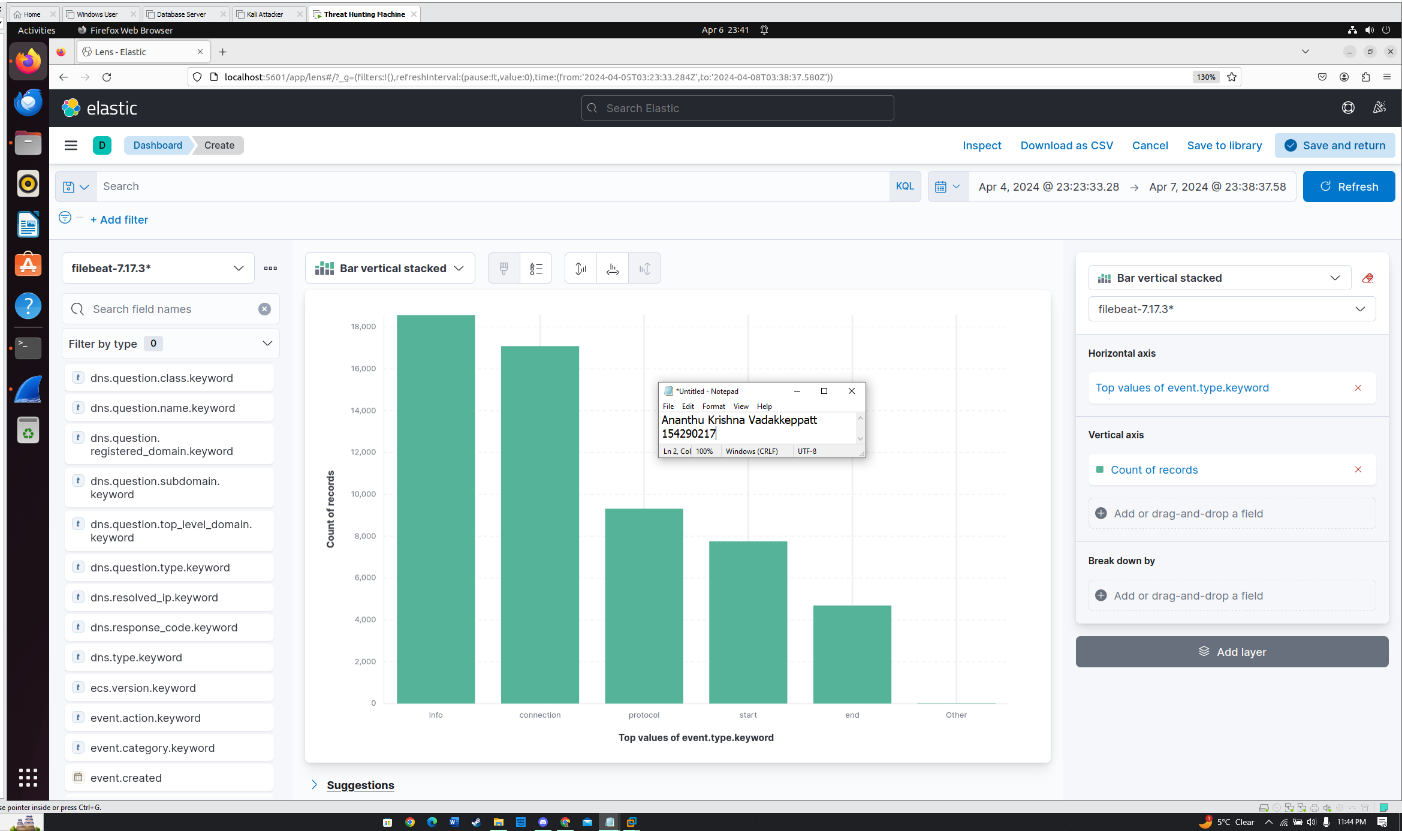
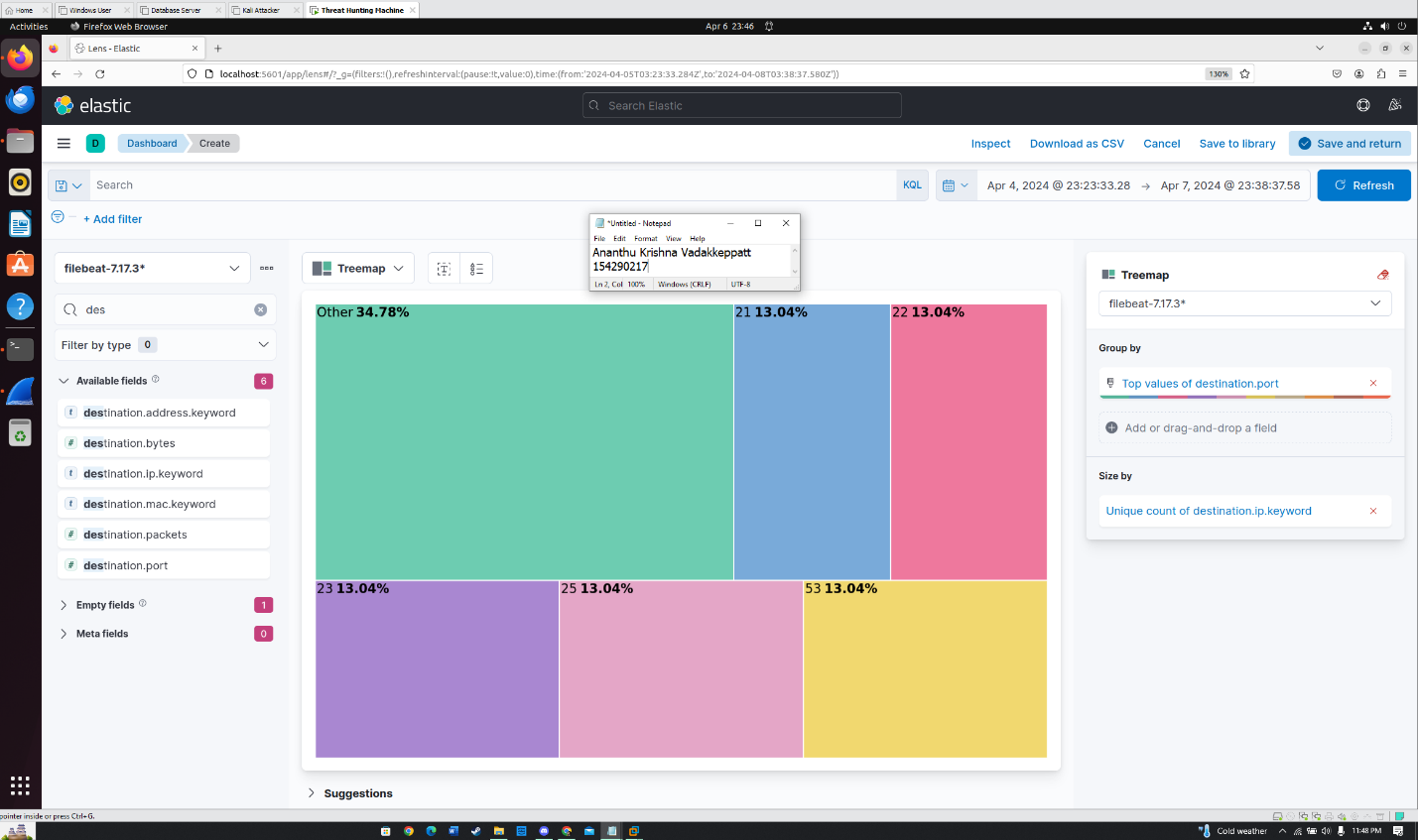
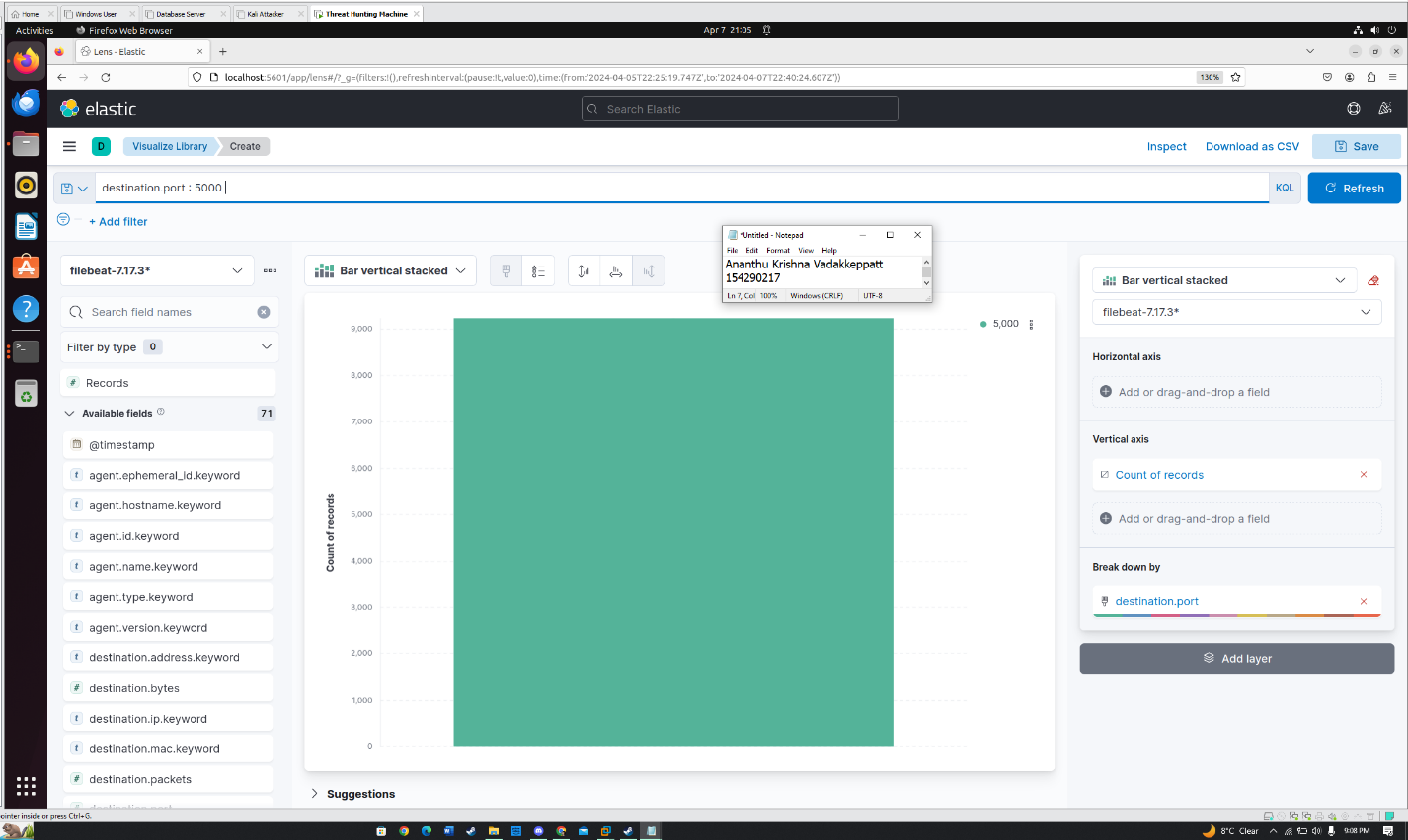
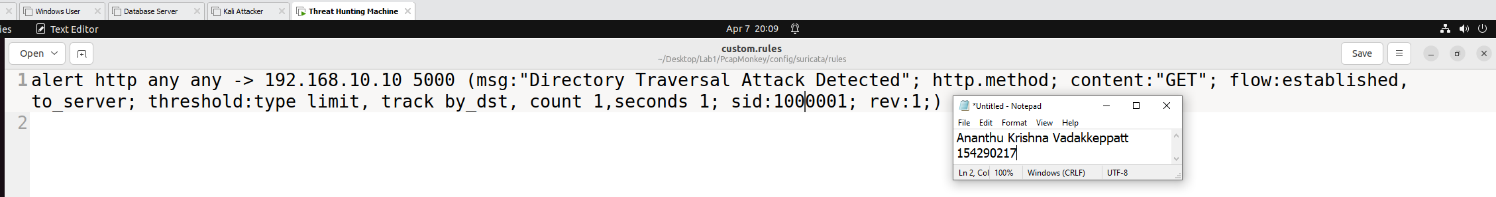


Fig.10 Zeek logs that are generated using pcapmonkey.

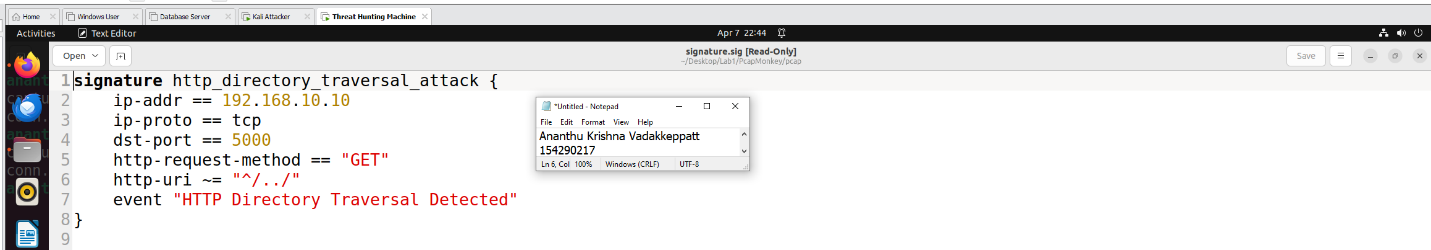
Fig.11 Visual created on Kibana that segregates the event types of across the entire Pcap.

Fig.12 Destination Ports Accessed on various IP Addresses.

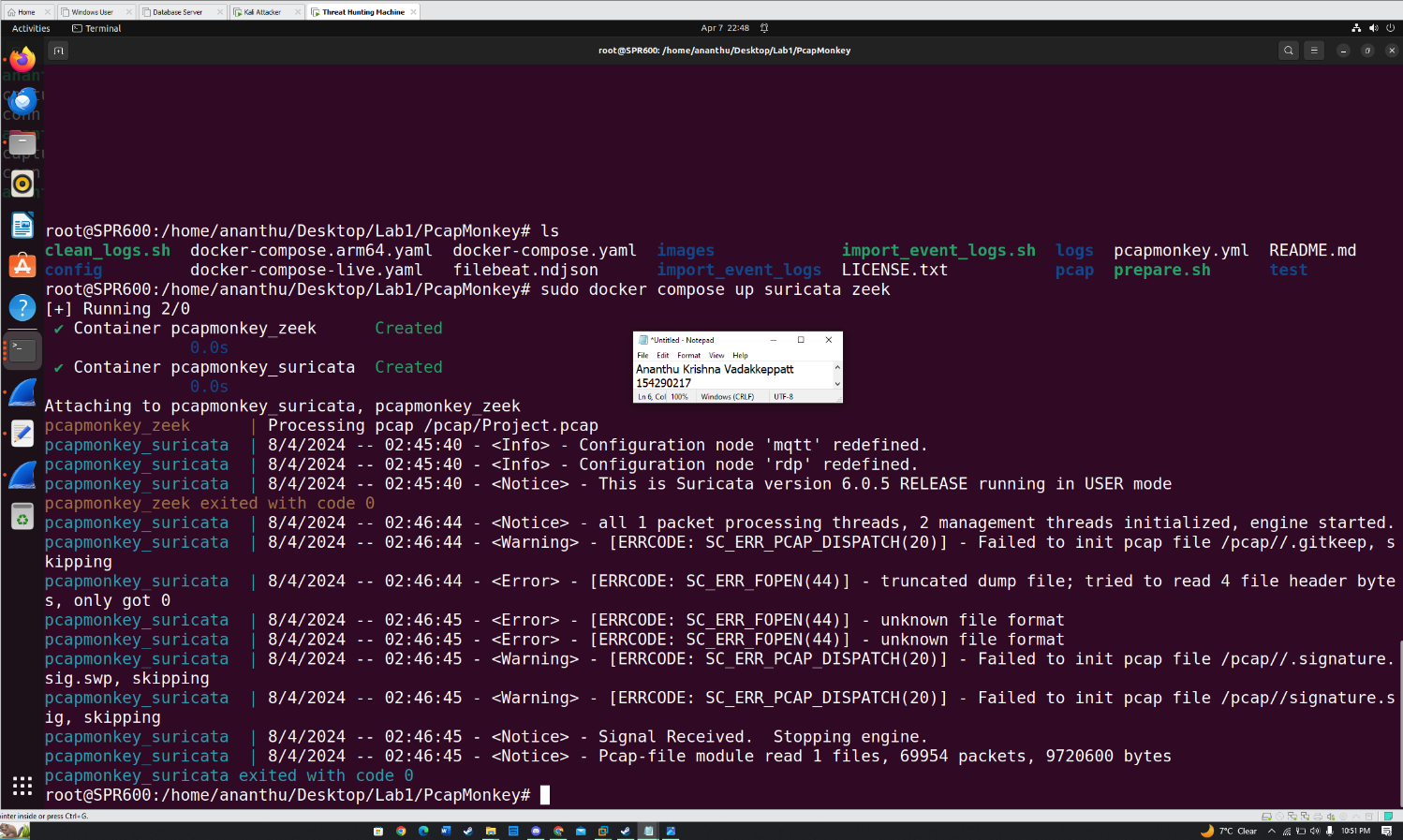
Fig.13 Number of packets that used the port 5000 which shows the intensity of the attack.

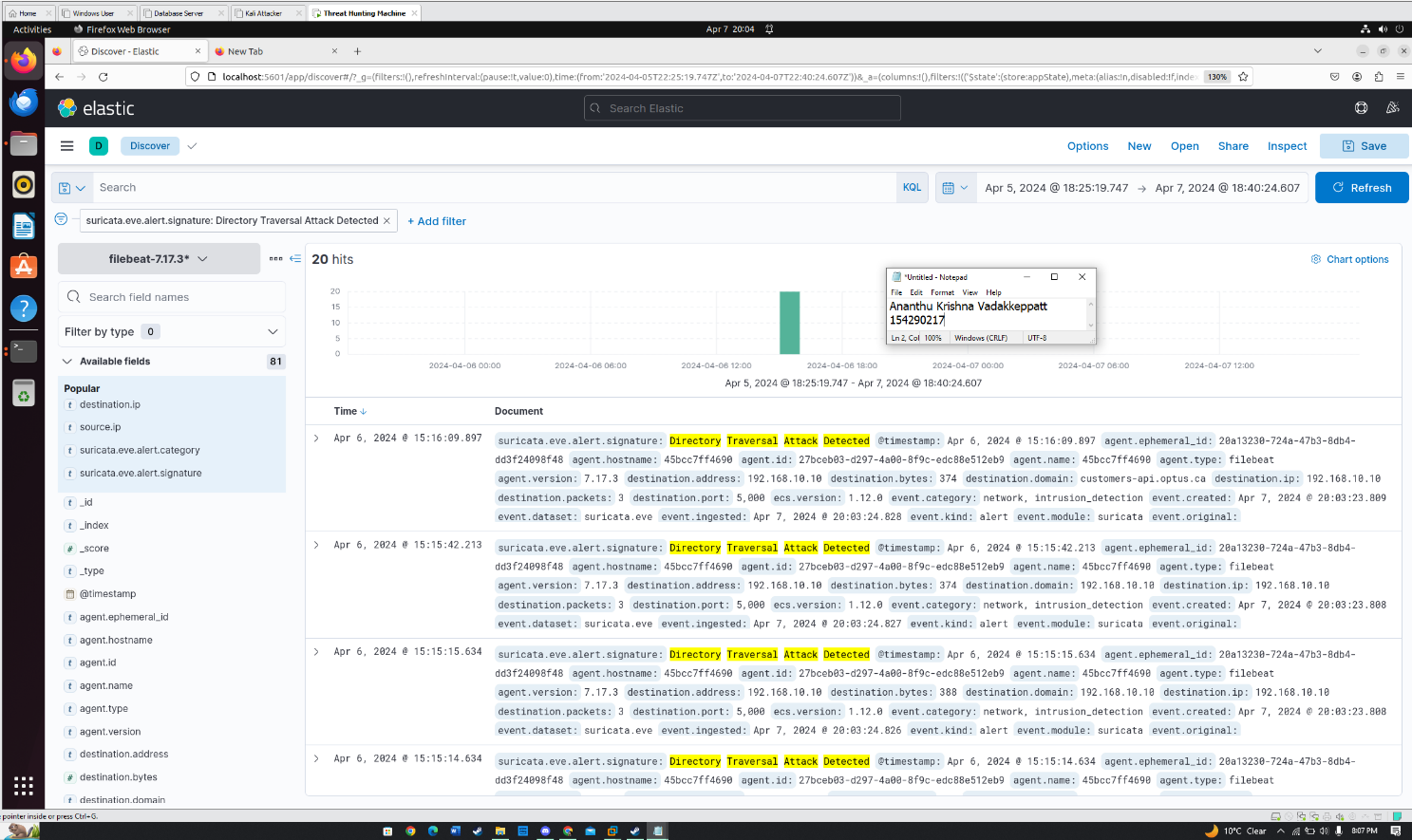
 Fig.14 Suricata rule created to detect Directory Traversal attacks on the web application.

This Suricata rule is designed to detect directory traversal attacks that were targeted on the open web application. Specifically targeted the GET requests along with attributes that focus on rate limiting.

Fig.15 Zeek Rule that is used to detect Directory Traversal attacks on the web application.

This Zeek rule focuses on identifying directory traversal attacks that are targeted on the API using different parameters shown in the above signature.

Fig.16 Loading up the signatures and creating an index.

 Fig.17 Verifying that the rule works on Kibana and flags the connections that seem to align with the requirements.

## Considerations and Limitations

While conducting an incident report there are various considerations and limitations that we would need to keep in mind. The incident report's objective is the priority, reminding the importance of making decisions based on facts uncovered from the investigation. Moreover, we should always ensure that our incident reports are detailed so that the reader can relate and agree with everything we have laid out across our discovery. We can encounter different limitations that may make it difficult for predicting the goal of an attacker. Incomplete information is one instance where it is difficult for someone to conclude. In such cases it is important to use other tools such as Zeek or ELK to confirm the verification of the results.

# **References**

1. *How did the Optus data breach happen? | UpGuard*. (n.d.-b). <https://www.upguard.com/blog/how-did-the-optus-data-breach-happen>
2. Da Veiga, M. S. (2021, December 9). Wireshark Filters List - Hacker Toolbelt - Medium. *Medium*. <https://medium.com/hacker-toolbelt/wireshark-filters-list-983c49468a45>
3. *8. Suricata Rules — Suricata 8.0.0-dev documentation*. (n.d.). <https://docs.suricata.io/en/latest/rules/index.html>
4. *Signature Framework — Book of Zeek (V6.2.0)*. (n.d.). <https://docs.zeek.org/en/current/frameworks/signatures.html>
5. *Introducing PCapMonkey*. (n.d.). Certego. <https://www.certego.net/blog/blog-3/>
6. Duncan, B. (2023, September 8). Wireshark tutorial: Display filter expressions. *Unit 42*. <https://unit42.paloaltonetworks.com/using-wireshark-display-filter-expressions/>