



Name: Caroline Wanjiku Macharia

Student ID: 669919

Lab4bInstructor: Professor Dennis Kaburu

Assignment: Lab 27.2.12 - Interpret HTTP and DNS Data to Isolate Threat Actor

Title: Advanced Information System Security

Course: ISC6120

Due: Summer 2025

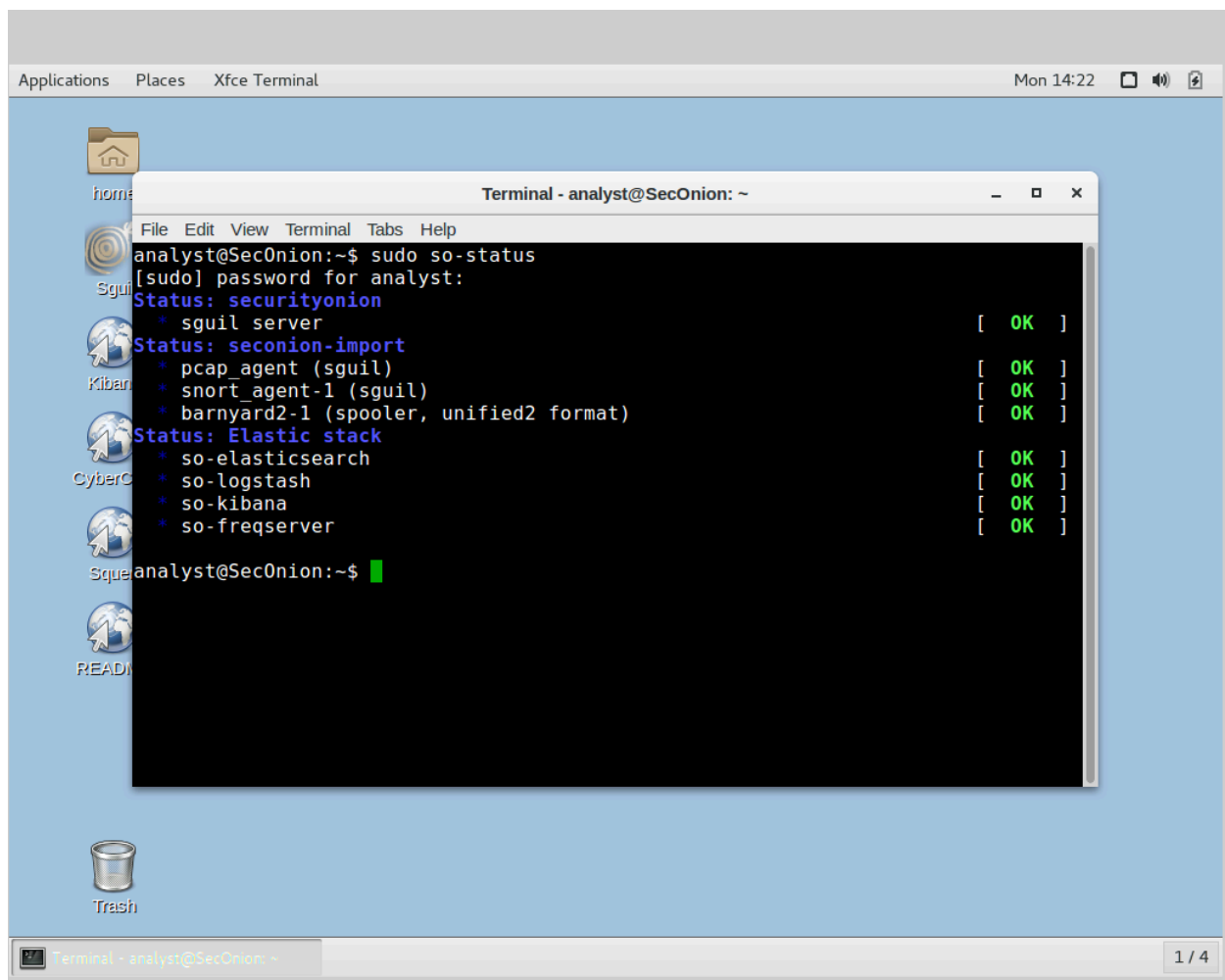
Part 1: Investigate an SQL Injection Attack

In this part, you will investigate an exploit in which unauthorized access was made to sensitive information that is stored on a web server. You will use Kibana to determine the source of the attack and the information accessed by the attacker.

Step 1: Change the time frame.

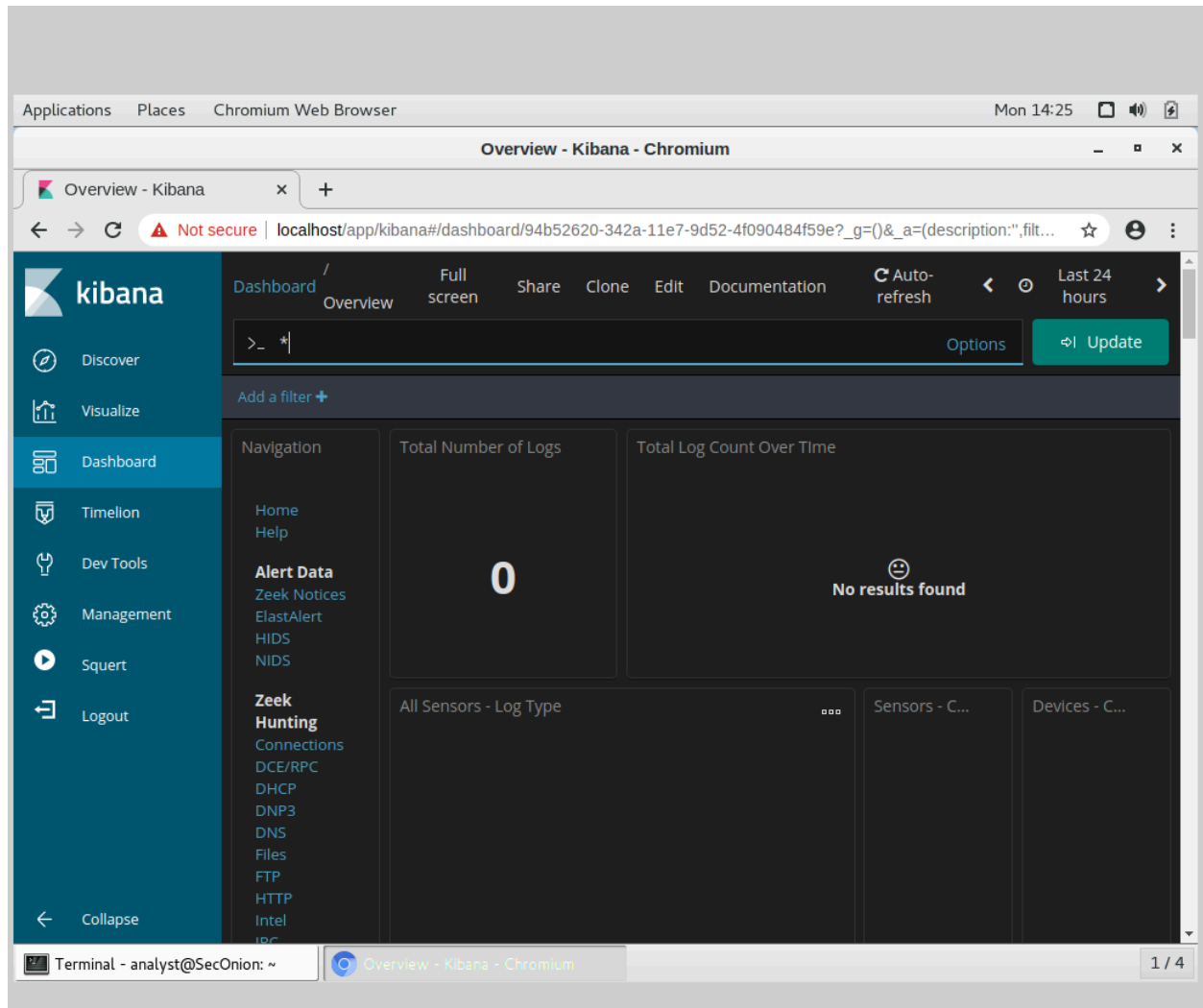
It has been determined that the exploit happened at some time during the month of June 2020. Kibana defaults to displaying data for the last 24 hours. You will need to change the time settings to see the data for the month of June 2020.

- Start the Security Onion VM and login with the username analyst and the password cyberops.
- Enter the `sudo so-status` command to check the status of services. The status for all the services should be OK before starting your analysis. This could take a few minutes.

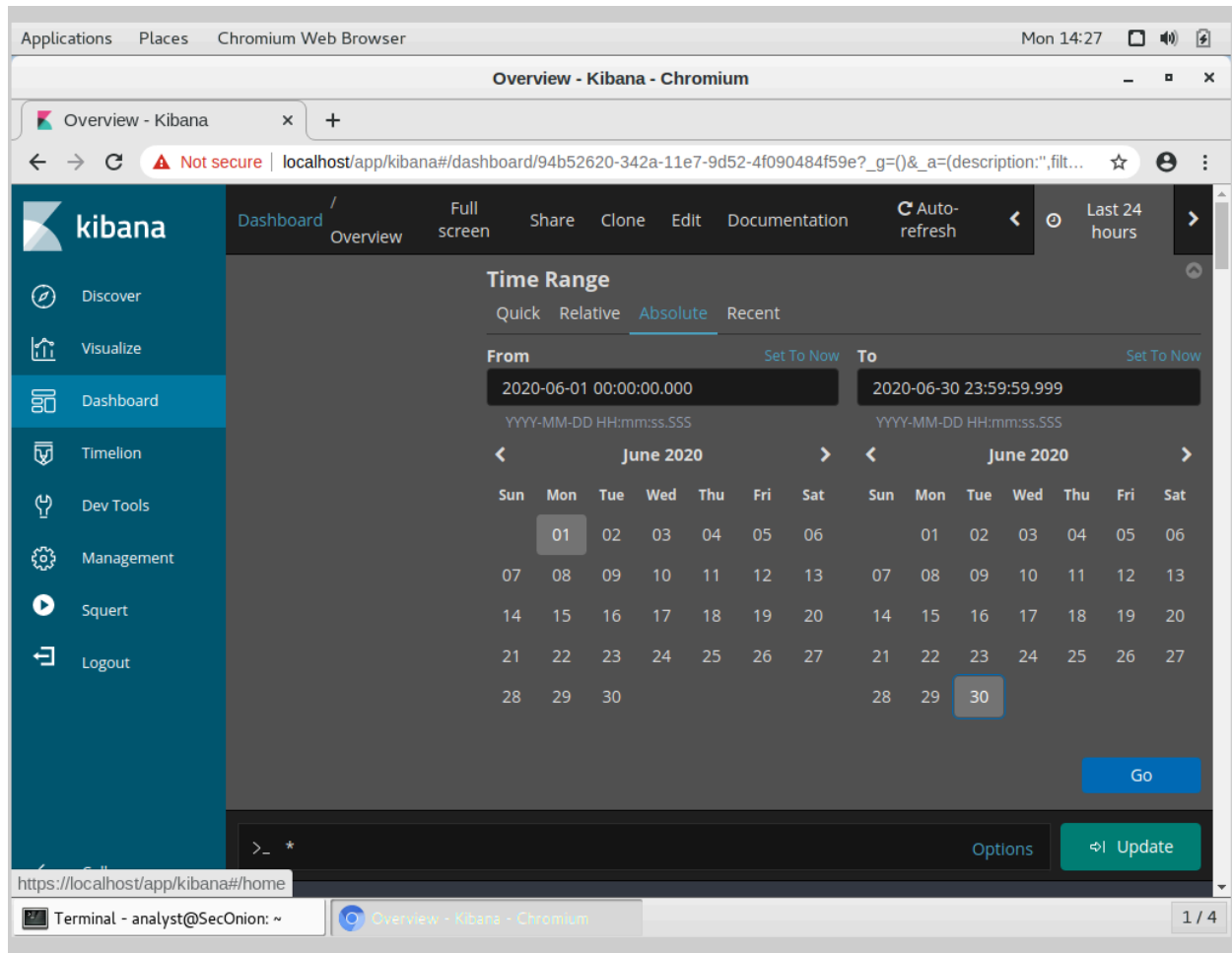


- After you log in, open Kibana using the shortcut on the Desktop. Login with the username analyst and the password cyberops. In Security Onion, Kibana has many pre-built dashboards and visualizations for monitoring and analysis. You can also create your own custom dashboards

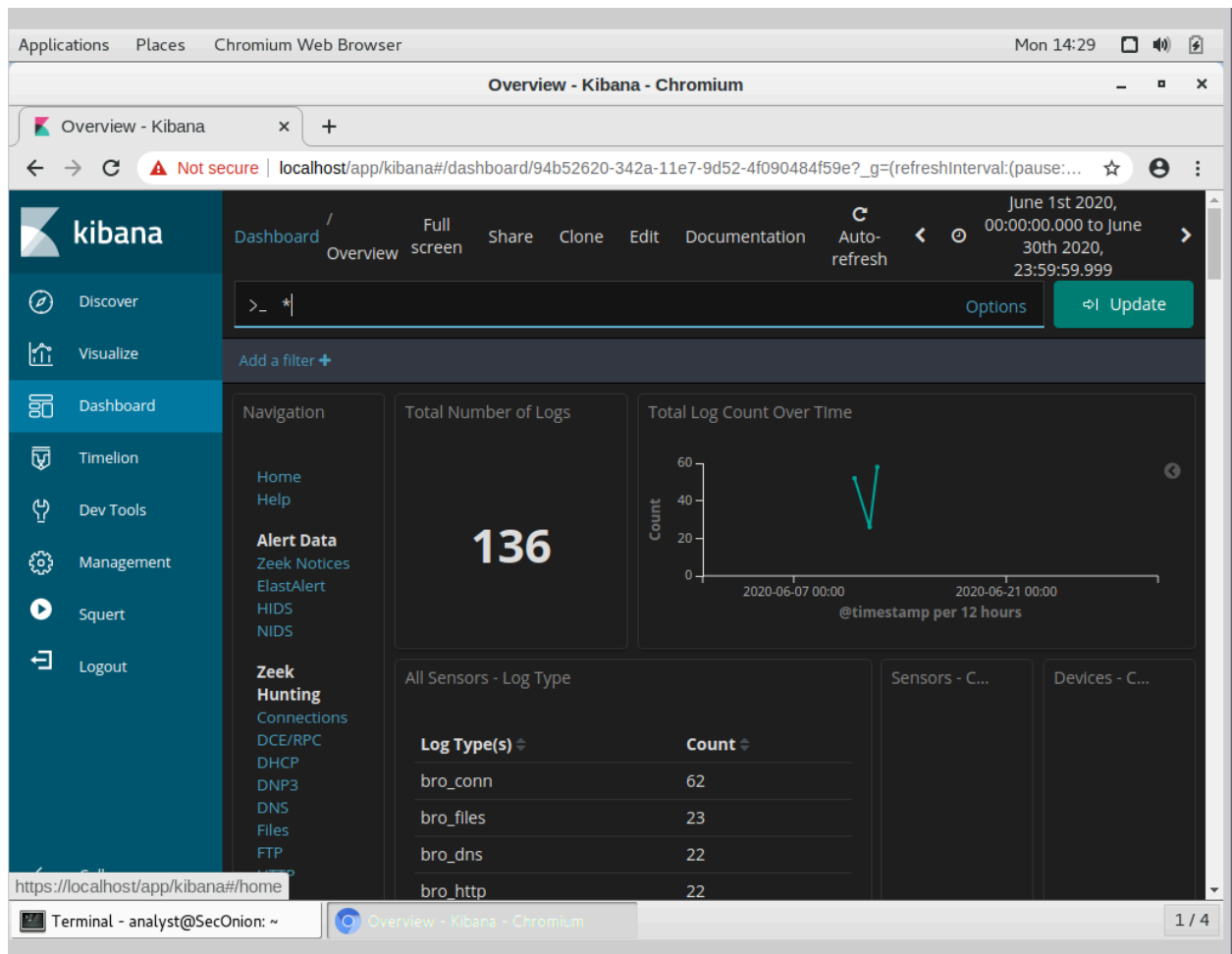
and visualizations catered to monitoring your particular network environment. Note: Your dashboard may not have any results in the last 24 hours.



d. In the upper-right corner of the window, click Last 24 hours to change the sample Time Range size. Expand the time range to include the interesting alerts. An SQL injection attack took place in June 2020 so that is what you need to target. Select Absolute under Time Range and edit the From and To times to include the entire month of June in 2020. Click Go to continue.

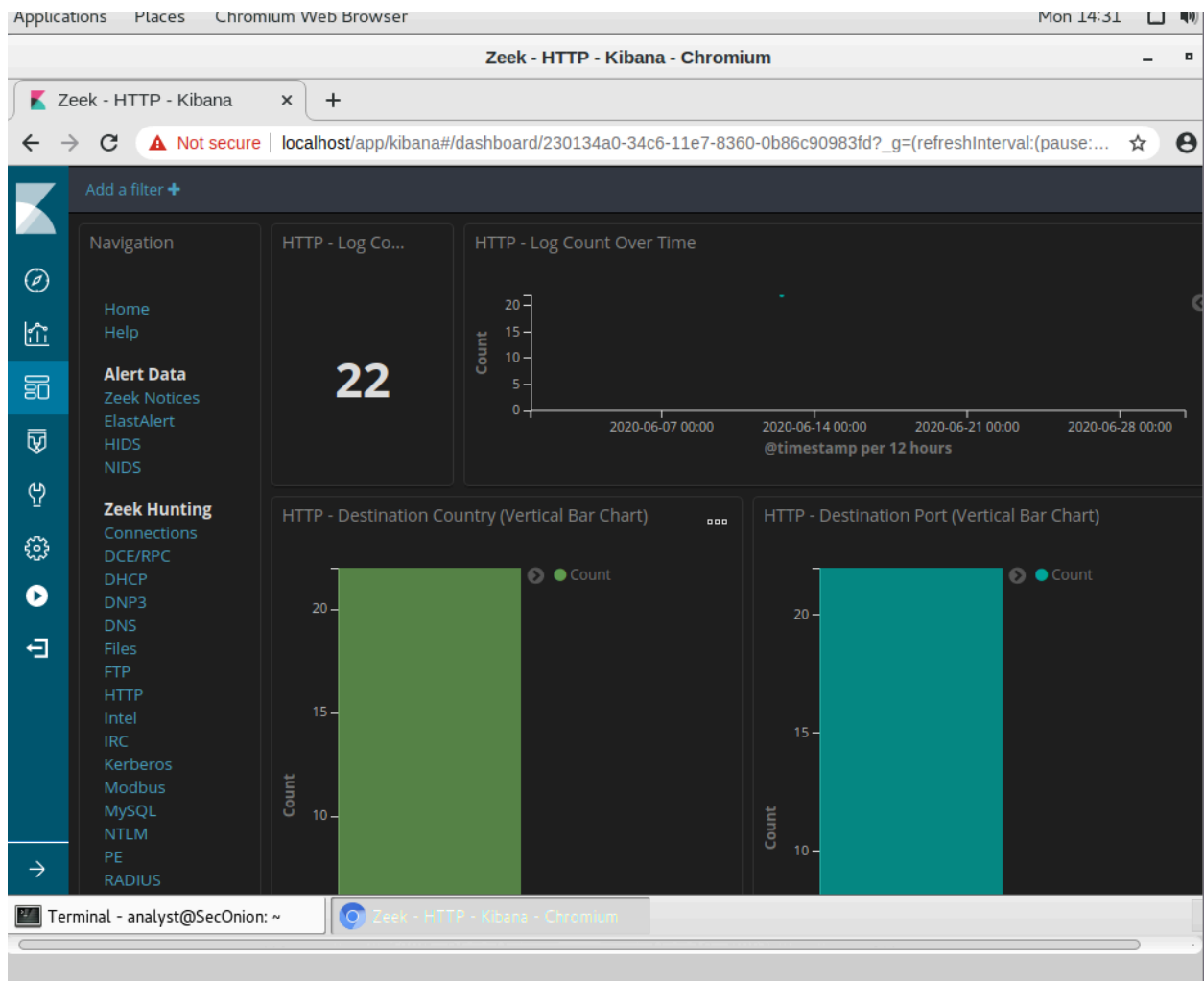


e. Notice the total number of logs for the entire month of June 2020. Your dashboard should be similar to that shown in the figure. Take a moment to explore the information that is provided by the Kibana interface.



Step 2: Filter for HTTP traffic.

a. Because the threat actor accessed data that is stored on a web server, the HTTP filter is used to select the logs associated with HTTP traffic. Select HTTP under the Zeek Hunting heading, as shown in the figure.



Scroll through the results and answer the following questions:

What is the source IP address?

209.165.200.227

What is the destination IP address?

209.165.200.235

What is the destination port number?

80

b. Scroll down to the HTTP Logs. The results list the first 10 results.

c. Expand the details of the first result by clicking the arrow that is next to the log entry timestamp. Note the information that is available.

Questions:

What is the timestamp of the first result?

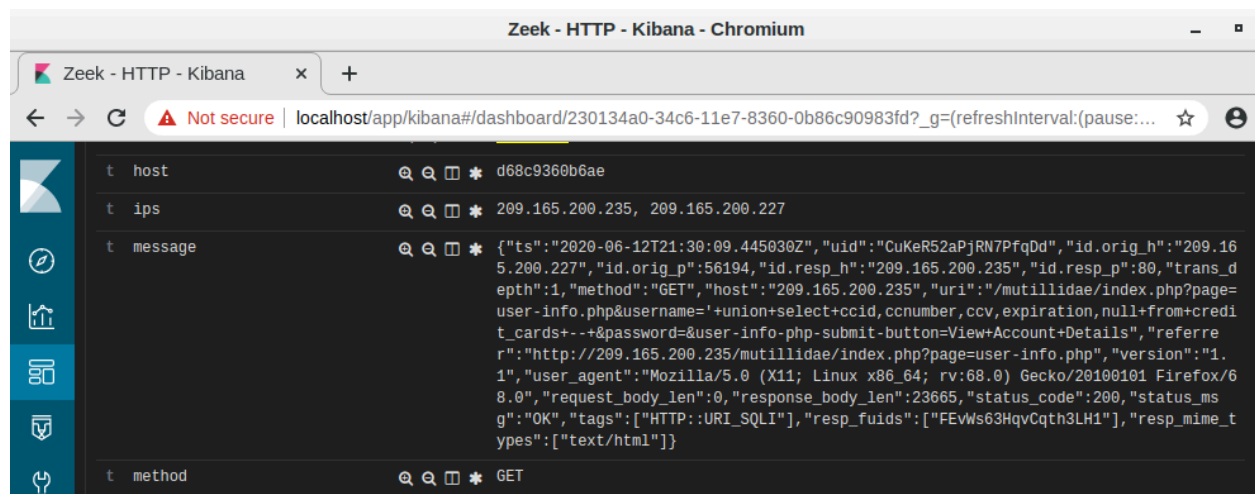
June 12th 2020, 21:30:09.445

What is the event type?

bro_http

What is included in the message field?

Response body, ts, uid, status_code, status_msg, method

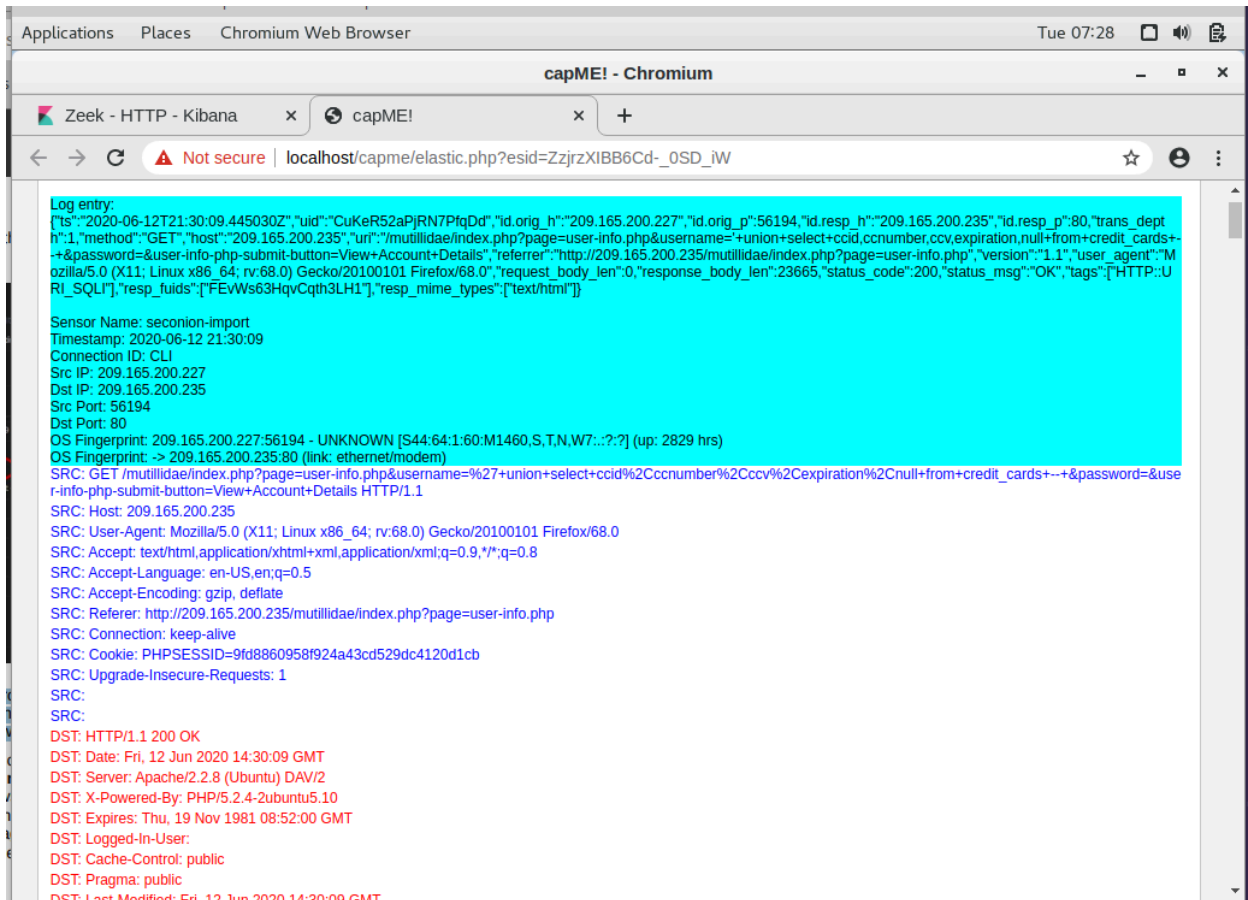


These are details about the HTTP GET request that was made by the client to the server. Focus especially on the uri field in the message text.

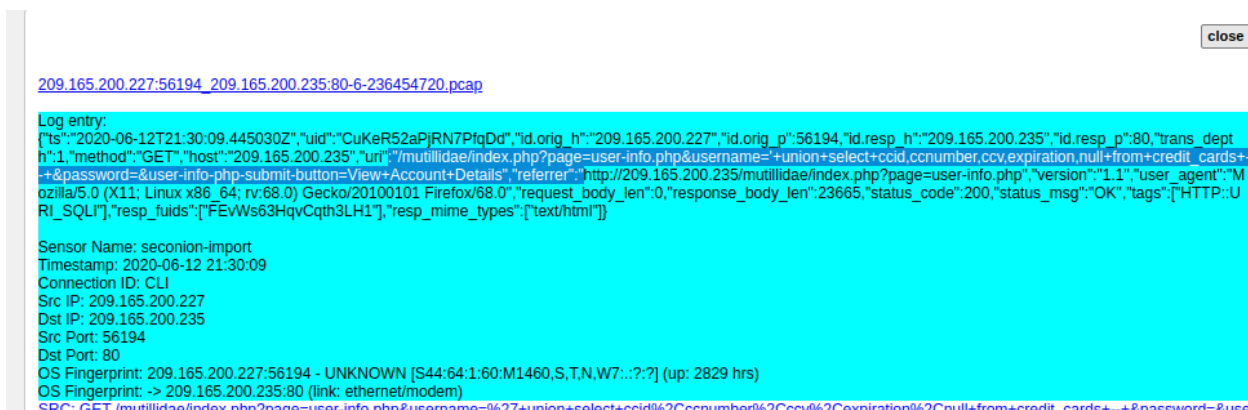
Step 3: Review the results.

a. Some of the information for the log entries is hyperlinked to other tools. Click the value in the alert_id field of the log entry to get a different view on the event.

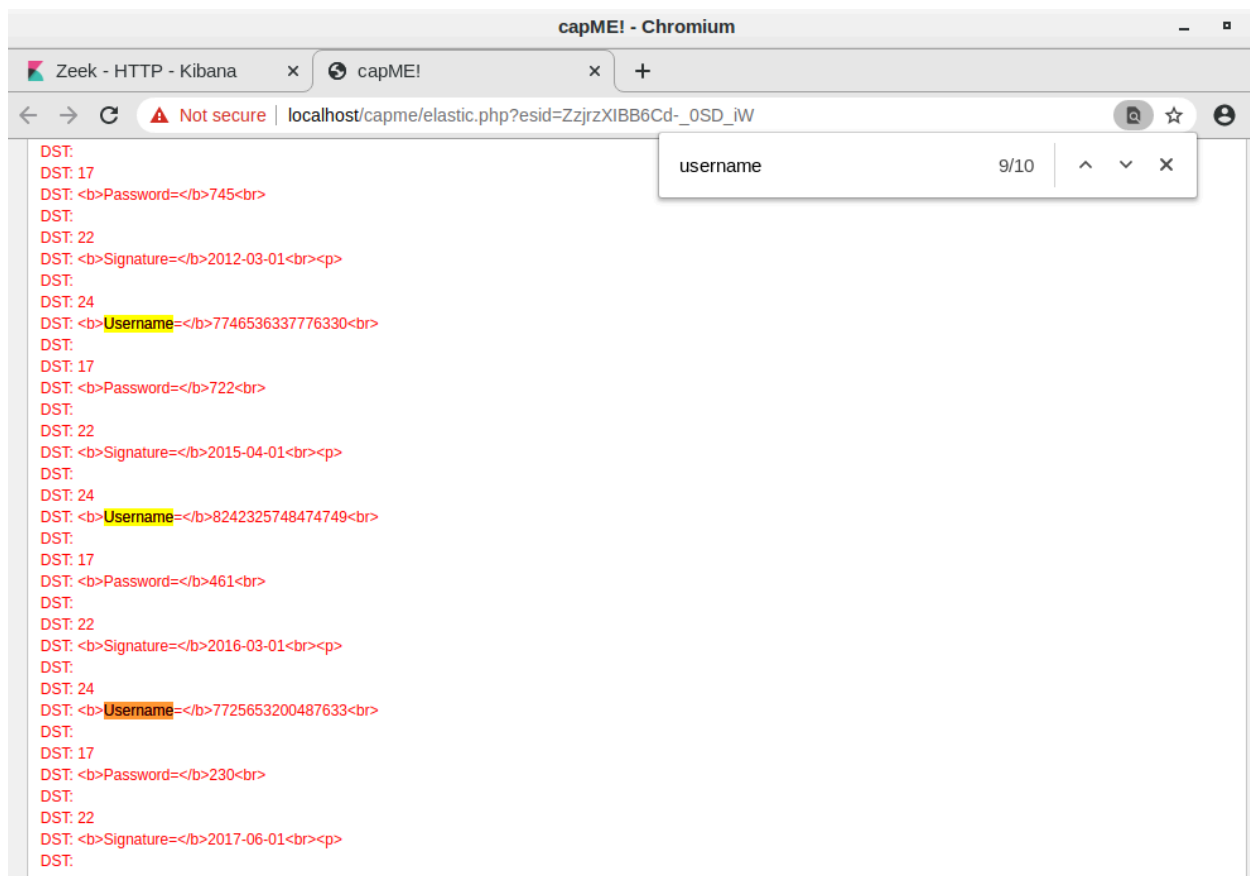
b. The result opens in a new web browser tab with information from capME!. capME! tab is a web interface that allows you to view a pcap transcript. The blue text contains HTTP requests that are sent from the source (SRC). The red text is responses from the destination web server (DST).



c. In the Log entry section, which is at the beginning of the transcript, notice the portion `username="+union+select+ccid,ccnumber,ccv,expiration,null+from+credit_cards+--+&password=` indicates that someone may have tried to attack the web browser using SQL injection to bypass authentication. The keywords, `union` and `select`, are commands that are used in searching for information in a SQL database. If the input boxes on a web page are not properly protected from illegal input, threat actors can inject SQL search strings or other code that can access data contained in databases that are linked to the web page



d. Find for the keyword username in the transcript. Use Ctrl-F to open a search box. Use the down arrow button in the search box to scroll through the occurrences that were found.



You can see where the term username was used in the web interface that is displayed to the user. However, if you look farther down, something unusual can be found.

Question: What do you see later in the transcript as regards usernames? There were usernames that were exfiltrated

Question: Give some examples of a username, password, and signature that was exfiltrated.



```
DST: 24
DST: <b>Username=</b>774653633776330<br>
DST:
DST: 17
DST: <b>Password=</b>722<br>
DST:
DST: 22
DST: <b>Signature=</b>2015-04-01<br><p>
DST:
```

```
DST: <b>Username=</b>8242325748474749<br>
DST:
DST: 17
DST: <b>Password=</b>461<br>
DST:
DST: 22
DST: <b>Signature=</b>2016-03-01<br><p>
DST:
```

```
DST: <b>Username=</b>7725653200487633<br>
DST:
DST: 17
DST: <b>Password=</b>230<br>
DST:
DST: 22
DST: <b>Signature=</b>2017-06-01<br><p>
DST:
DST: 24
DST: <b>Username=</b>1234567812345678<br>
DST:
DST: 17
DST: <b>Password=</b>627<br>
DST:
DST: 22
DST: <b>Signature=</b>2018-11-01<br><p>
```

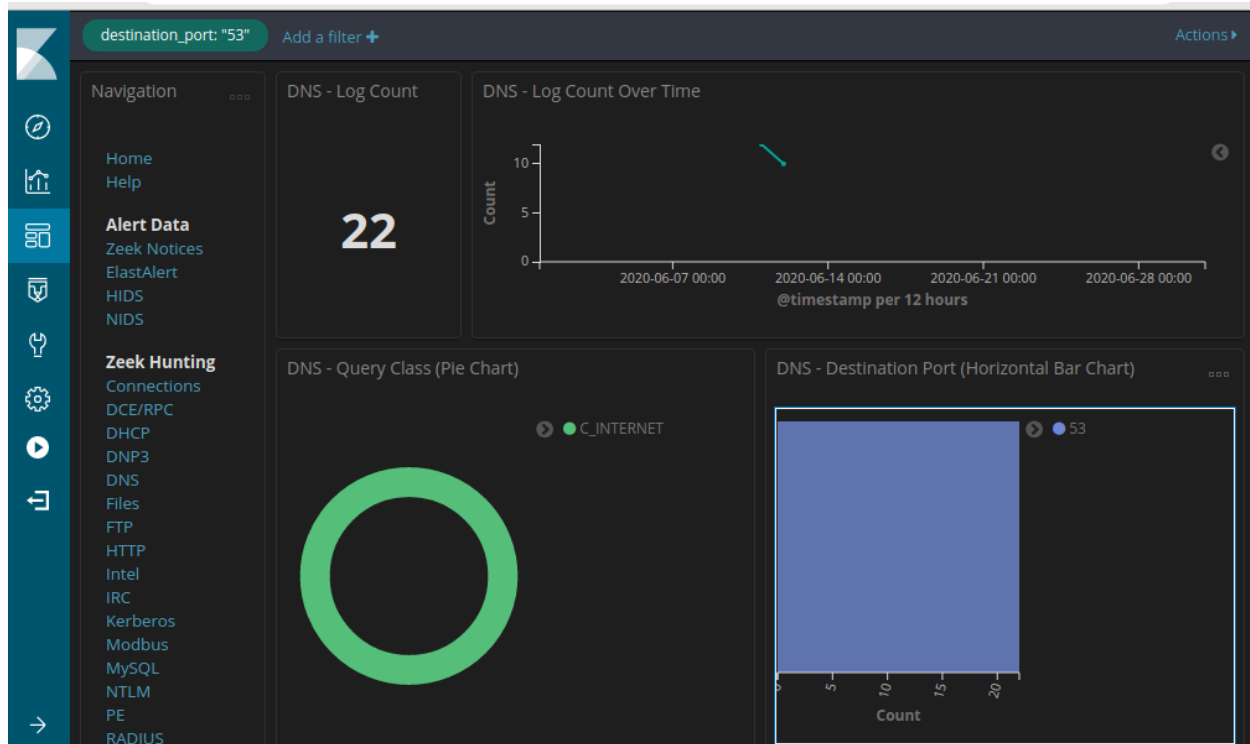
e. Close the capME! tab and return to Kibana.

Part 2: Analyze DNS exfiltration.

A network administrator has noticed abnormally long DNS queries with strange looking subdomains. Your job is to investigate the anomaly.

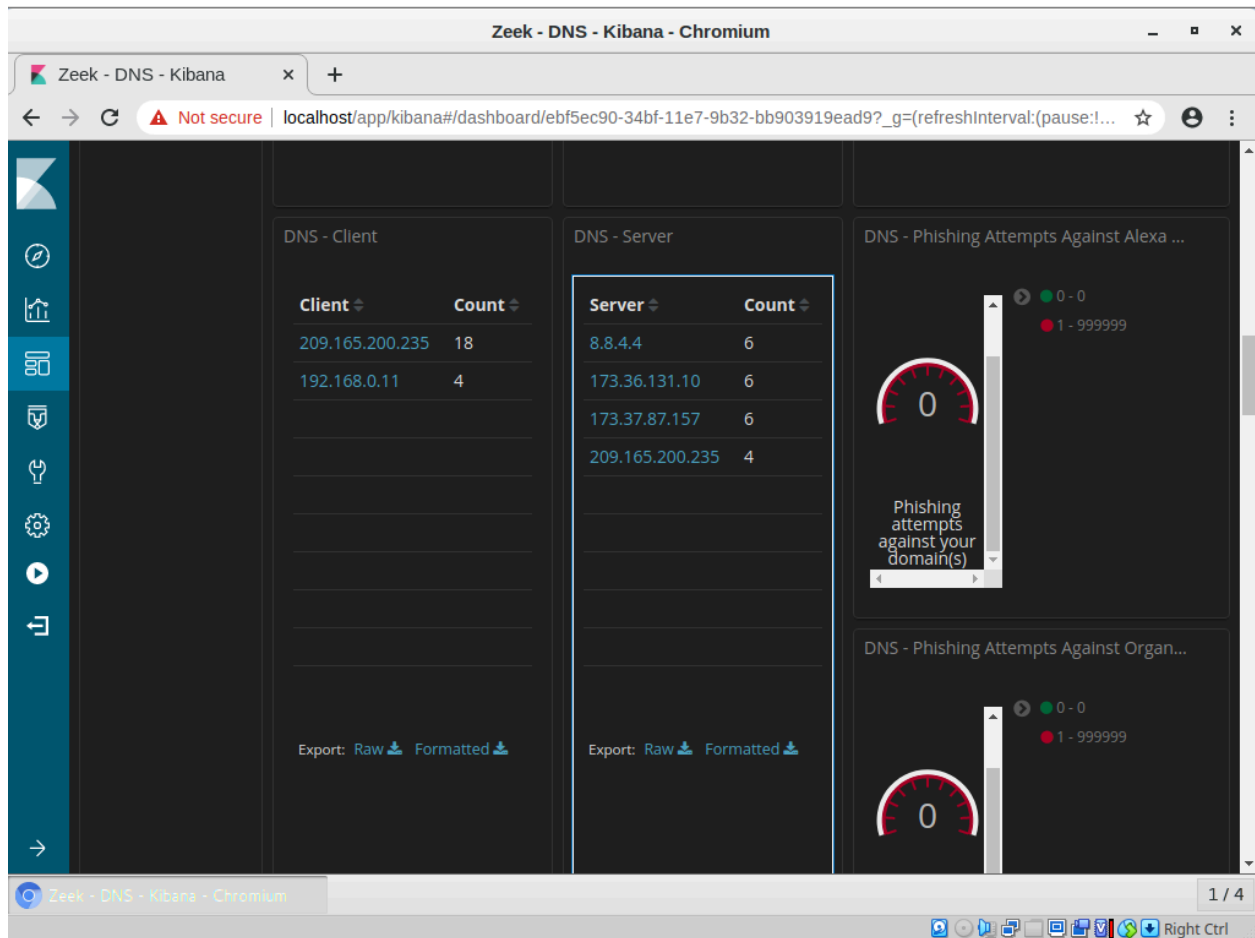
Step 1: Filter for DNS traffic.

- From the top of the Kibana Dashboard, clear any filters and search terms and click Home under the Navigation section of the Dashboard. The Time period should still include June 2020.
- In the same area of the Dashboard, click DNS in the Zeek Hunting section. Notice the DNS Log Count metrics and Destination Port horizontal bar chart.

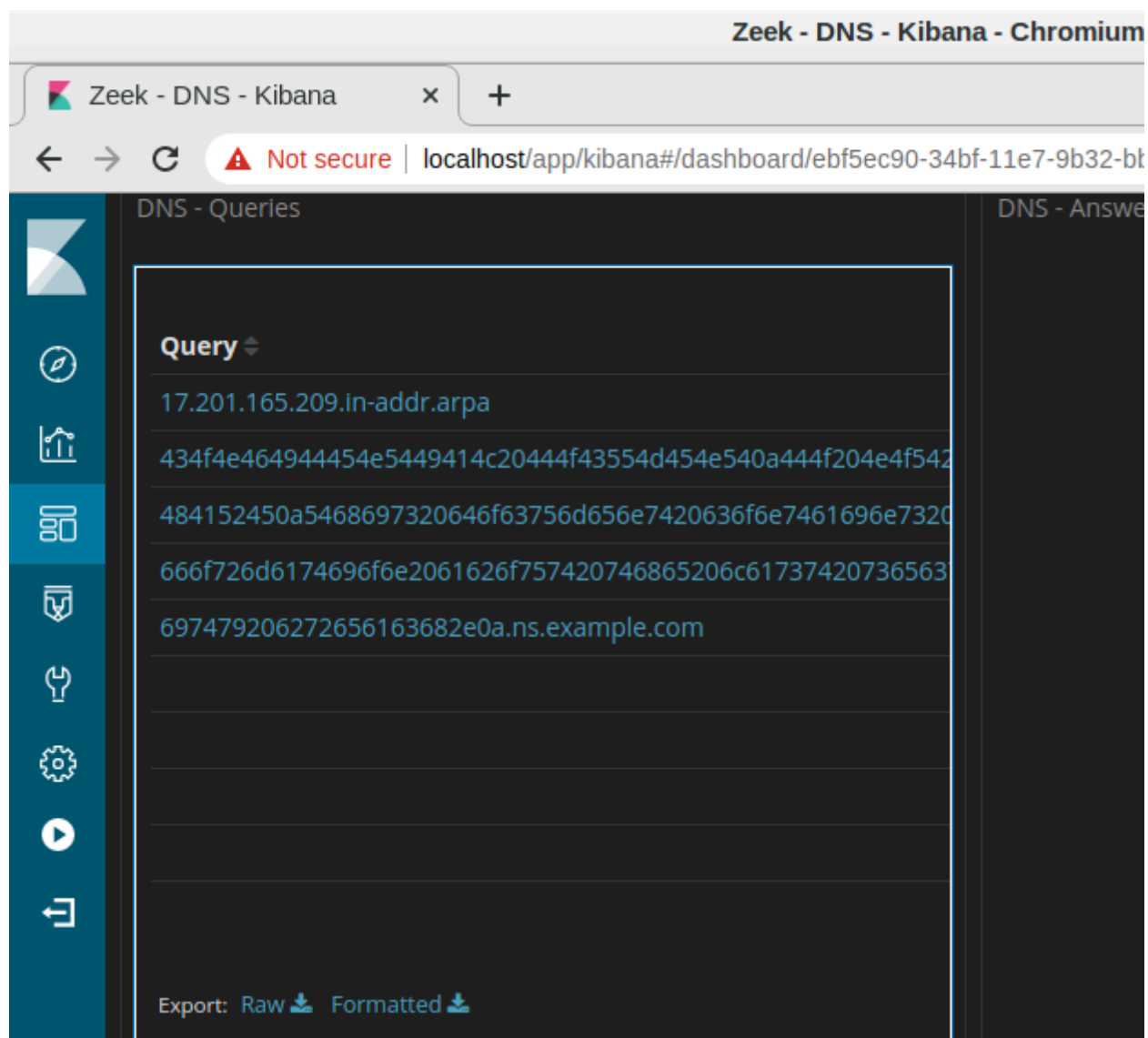


Step 2: Review the DNS-related entries.

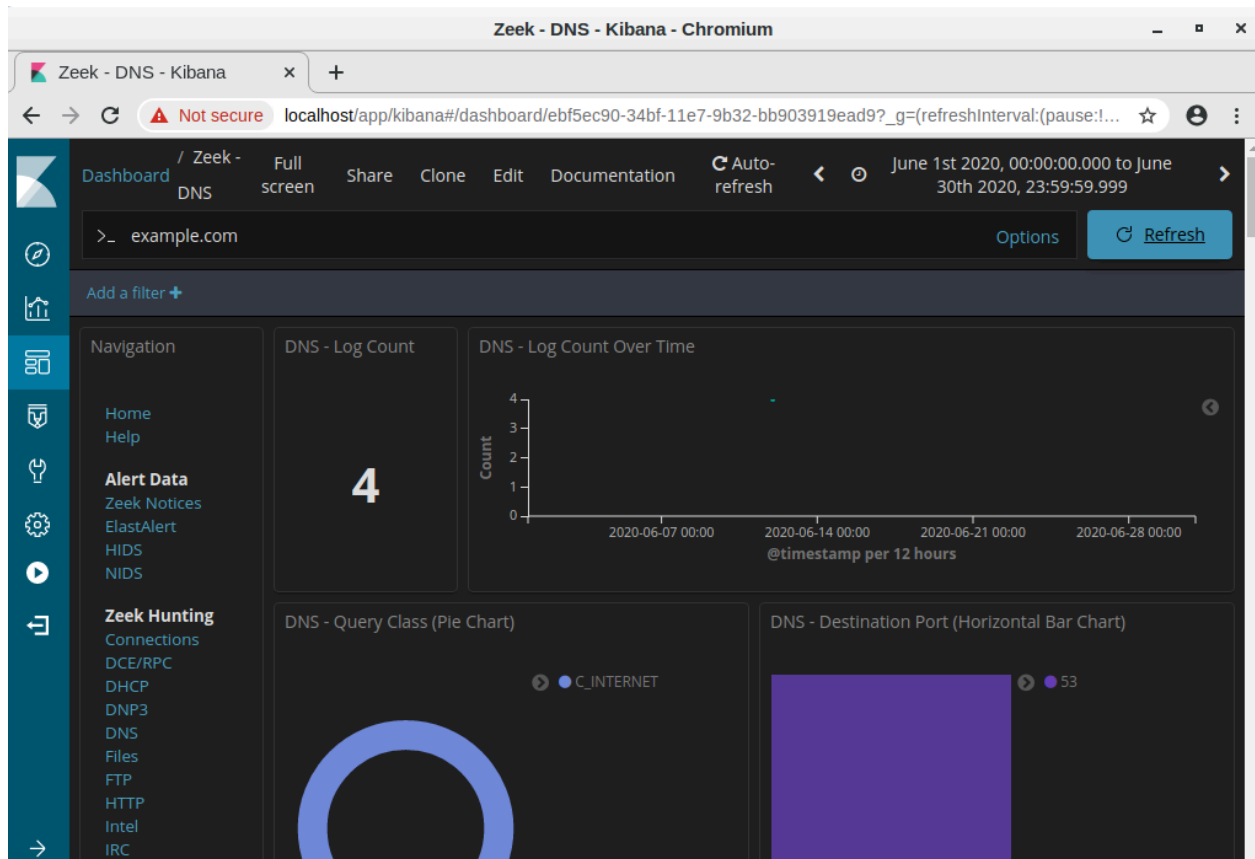
- Scroll down the window. You can see the top DNS query types. You may see address records (A record), IPv6 address Quad A records (AAAA), NetBIOS records (NB) and a pointer records for resolving the hostnames (PTR). You can also see the DNS response codes.
- By Scrolling further down, you can see a list of the top DNS clients and DNS Servers based on their request and response counts. There is also a metric for number of DNS Phishing attempts, which are also known as DNS pharming, spoofing, or poisoning.



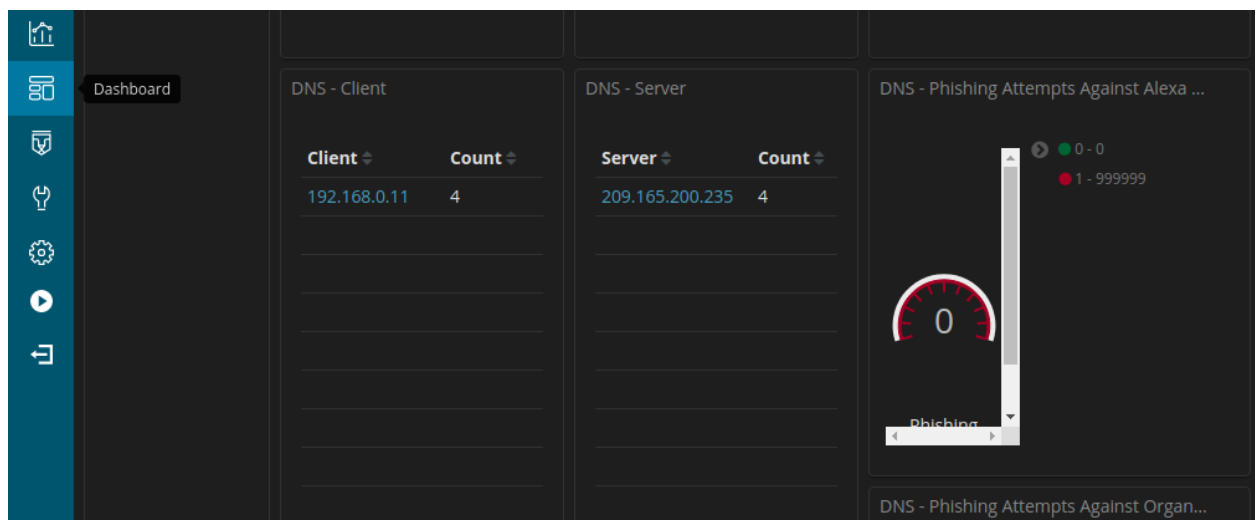
c. Scrolling further down the window you can see a listing of the top DNS queries by domain name. Notice how some of the queries have unusually long subdomains attached to ns.example.com. The domain example.com should be investigated further.



d. Scroll back to the top of the window and enter example.com in the search bar to filter for example.com and click Update. Note that the number of entries in the Log Count is smaller because the display is now limited to requests to the example.com server.



e. Locate information about the DNS - Client and DNS - Server.

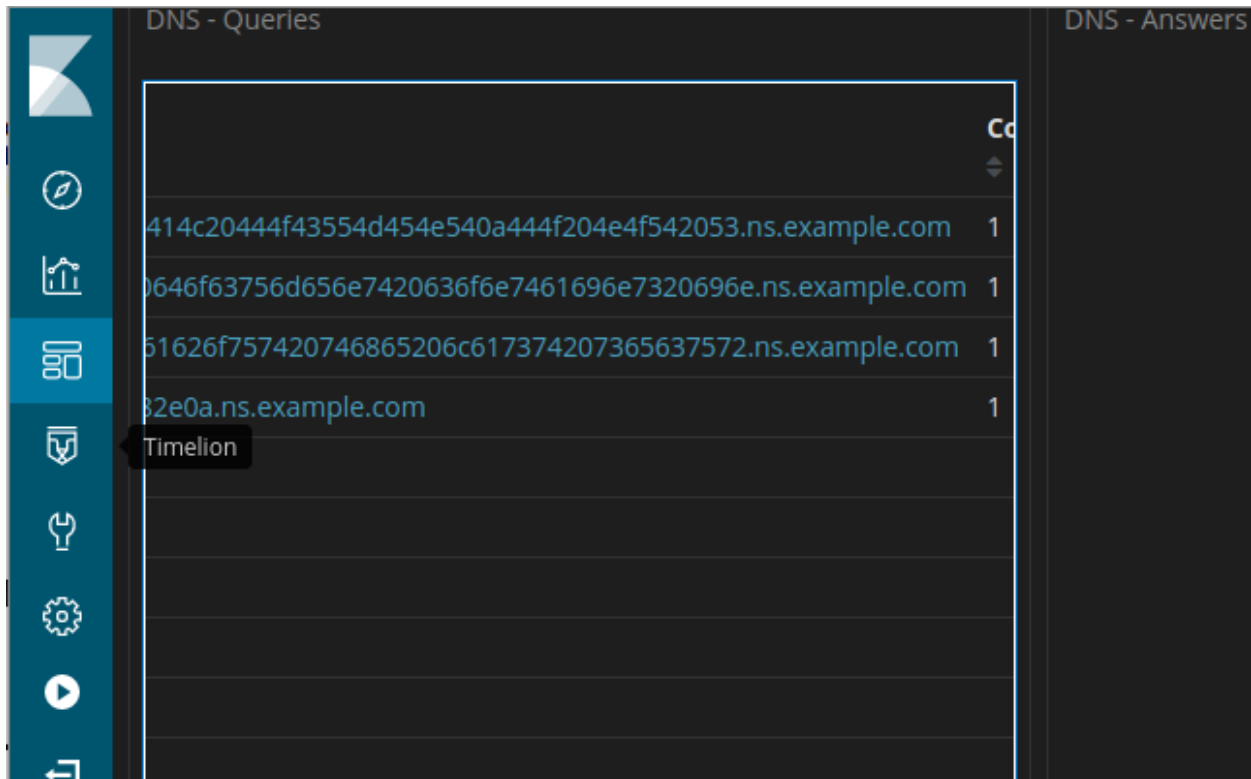


Record the IP addresses of DNS client and server.

DNS - Client : 192.168.0.11 DNS-Server : 209.165.200.235

Step 3: Determine the exfiltrated data.

a. Continue to scroll further down to see four unique log entries for DNS queries to example.com. Notice how the queries are to suspiciously long subdomains attached to ns.example.com. The long strings of numbers and letters in the subdomains look like text encoded into hexadecimal (0-9, a-f) rather than legitimate subdomain names. Click the Export: Raw download link to download the queries to an external file. A CSV file is downloaded to the /home/analyst/Downloads folder.

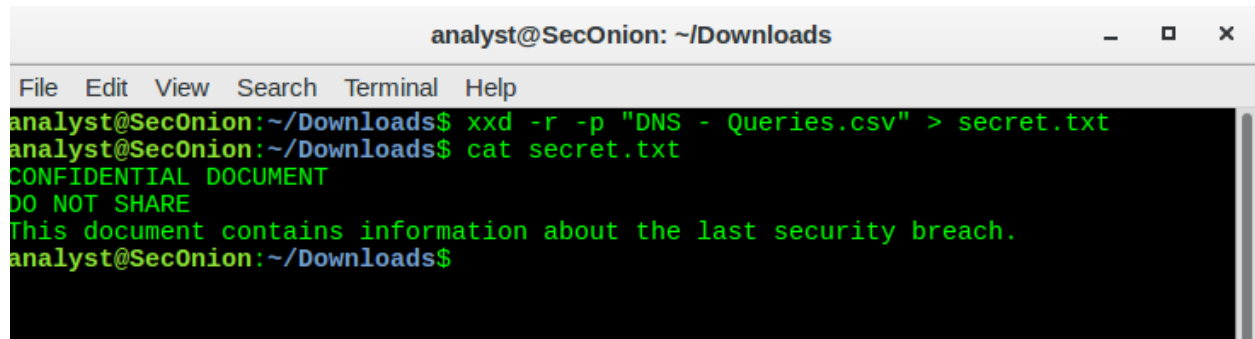


b. Navigate to the /home/analyst/Downloads folder. Open the file using a text editor, such as gedit. Edit the file by deleting the text surrounding the hexadecimal portion of the subdomains, leaving only the hexadecimal characters. Be sure to remove the quotes too. The contents of your file should look like the information below. Save the edited text file with the original file name.



c. In a terminal, use the xxd command to decode the text in the CSV file and save it to a file named secret.txt. Use cat to output the contents of secret.txt to the console.

Were the subdomains from the DNS queries subdomains? If not, what is the text?

A terminal window titled 'analyst@SecOnion: ~/Downloads' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
analyst@SecOnion:~/Downloads$ xxd -r -p "DNS - Queries.csv" > secret.txt
analyst@SecOnion:~/Downloads$ cat secret.txt
CONFIDENTIAL DOCUMENT
DO NOT SHARE
This document contains information about the last security breach.
analyst@SecOnion:~/Downloads$
```

What does this result imply about these particular DNS requests? What is the larger significance?

The results indicate that the DNS requests were separate, coordinated requests containing hidden content. The larger significance of the result is that DNS queries could be used to hide the sending of files and bypass network security

What may have created these encoded DNS queries and why was DNS selected as the means to exfiltrate data?

They might have been created by a malware by cycling through documents on the host and encoding their contents in hexadecimal and then creating DNS queries that use the hexadecimal strings as DNS subdomains. DNS requests are very commonly sent out of a network to the internet, so DNS requests may not be monitored.