Logo

Description automatically generated

**Andre Rodrigues Russo**

[Rodriguesrusso.a@northeastern.edu](mailto:Rodriguesrusso.a@northeastern.edu)

**Anukriti Baijal**

[Baijal.a@northeastern.edu](mailto:Baijal.a@northeastern.edu)

**Sarangapani Balaji Manideep Sai Aellam**

[Aellam.s@northeastern.edu](mailto:Aellam.s@northeastern.edu)

**Trilok Palla**

[Palla.t@northeastern.edu](mailto:Palla.t@northeastern.edu)

**Northeastern University**College of Professional Studies

**ALY6110:** Data Management and Big Data  
**CRN:** 80437

**ANALYSIS OF NETWORK ATTACKS**

**Professor:** Valeriy Shevchenko

**May 18th, 2023**

**Dataset Description: UNSW-NB15**

The IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) developed the raw network packets for the UNSW-NB15 dataset in order to provide a hybrid of real contemporary normal activities and synthetic current attack behaviors.

Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine attack categories in this data collection. To create a total of 49 features with the class label, twelve algorithms are built, and the Argus and Bro-IDS tools are employed.

The overall size is approximately 600MB, which is sufficient for using big data analytics techniques.

**Real-world Problem Description:**

Network security-related challenges including intrusion detection, network traffic analysis, and cyber-attacks can all be resolved using the UNSW-NB15 dataset. The UNSW-NB15 dataset can be used to address a practical issue by identifying the most frequent attack types and their features. This can assist organizations in prioritizing their security efforts and developing focused protection tactics against different sorts of assaults.

To evaluate the dataset and determine the top three most frequent attack types and their associated characteristics, such as time, protocol type, service, flag, and others, an organization could, for instance, utilize Hive queries or SQL queries in Databricks. The company can utilize this knowledge to create a set of regulations and guidelines that are tailored to these kinds of assaults.

Additionally, trends and anomalies in network traffic that can point to a potential security risk can be found using dataset insight searches. Organizations can spot suspicious trends in the network traffic statistics, such as a sudden surge in traffic volume or unusually high traffic coming from a certain IP address. The development of proactive defense tactics and the early detection of security breaches are therefore possible using this knowledge.

Using big data analytics and insight queries, the UNSW-NB15 dataset may be utilized to learn important things about network security risks and create strong protection methods.

**Methodology utilized to analyze dataset:**

* **Cloudera, Hadoop, and Hive:**

**Prepare the data:** Before you can start analyzing the data, you need to prepare it for ingestion into the Hadoop cluster. This may involve cleaning the data, transforming it into a suitable format, and splitting it into smaller files if necessary.

**Load the data into Hadoop:** Once the data is prepared, you can load it into the Hadoop Distributed File System (HDFS). You can also use the Hadoop command-line interface to load the data directly into HDFS.

**Create a Hive database:** To analyze the data using Hive, you need to create a Hive database and define a schema for the data. The schema should include the column names and data types for each field in the dataset.

**Create external tables:** After creating the database, you can create external tables that point to the data stored in HDFS. You can use the CREATE EXTERNAL TABLE statement in Hive to define the table schema and specify the location of the data in HDFS.

**Hive Insight queries:** With the tables in place, you can now use Hive queries to analyze the data. Hive provides a SQL-like interface for querying the data and supports a wide range of built-in functions and operators.

**Visualize the data:** Finally, you can use Hive visualizations such as charts, graphs, and maps to visualize the results of your queries.

* **Databricks**:

**Data Preparation:** This is like a data preparation process for Hadoop. The needs to be clean, formatted and reduced into different files, if needed.

**Load the Data into Databricks**: To load the data into Databricks File System (DBFS), you can use the simple UI that simply imports the data into the system.

**Creating a Cluster**: To be able to perform different operations and analysis on the data, a cluster needs to be created and applied in the notebook. This can be done by simply using the option provided in the upper-right corner of the notebook.

**Create table**: Once a cluster has been created, the data needs to be copied into a table that will be utilized during the analysis. Databricks allows 4 different languages- R, Scala, Python and SQL. You can use either of these languages to copy data into the table. In this project, SQL was utilized.

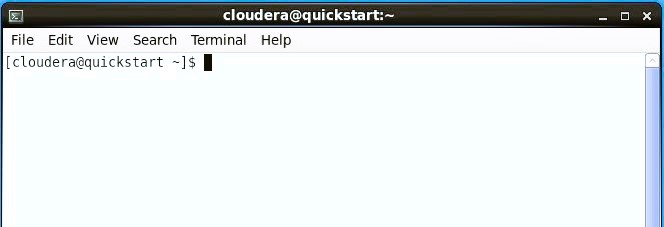
**Insight Queries**: Once the tables are set up, you may analyze the data with SQL queries, Python, R, or Scala commands. With a variety of built-in functions and operators, Databricks supports several languages and offers a versatile interface for data querying.

**Steps taken to analyze the data:**

Data Analysis is the process of extracting insights and knowledge from large datasets, and it is a critical component of big data analytics. However, with the exponential growth of data, traditional data mining techniques are no longer sufficient. Hive is defined as a data warehousing and SQL-like query language tool built on top of Hadoop, that is used to provide a powerful platform for analyzing big data. Another technology that is very useful is Databricks. Databricks is a platform for data engineering and analytics built on Apache Spark. It allows users to use multiple languages—R, Scala, Python and SQL, to analyze the data. In this documentation, we will explore how to use Hive and Databricks for data mining and uncover valuable insights from large datasets.

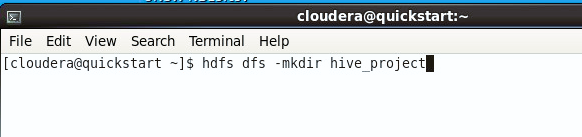
* **Hive:**

**Here are the steps with screen shots of queries and analysis using Hive:**

**Step 1**: Open terminal in the Cloudera VM.

**Step 2:** In this step the command "hdfs dfs -mkdir hive\_project" is used to create a new directory named "hive\_project" in HDFS (Hadoop Distributed File System).

Assuming you have the necessary permissions to create a new directory in HDFS, running this command will create a new directory named "hive\_project" in the current working directory of HDFS. If the current working directory does not exist, the command will create it first and then create the "hive\_project" directory inside it.

****Once the directory is created, you can use it to store data or create a Hive table using Hadoop and the Hive data warehouse tool.

**Step 3:** The "pwd" command stands for "print working directory". When you run this command in a terminal or command prompt, it will display the current working directory, which is the directory you are currently located in.

The command "hdfs dfs -put /home/cloudera/Desktop/UNSW-NB15.csv /user/cloudera/hive\_project" is used to copy a file named "UNSW-NB15.csv" from the local file system to HDFS.

Assuming you have the necessary permissions to write to the "/user/cloudera/hive\_project" directory in HDFS, running this command will copy the "UNSW-NB15.csv" file from the "/home/cloudera/Desktop" directory on your local file system to the "/user/cloudera/hive\_project" directory in HDFS.

Once the file is copied, you can use it as a data source for your Hadoop and Hive projects.

**hdfs dfs -put /home/cloudera/Desktop/UNSW-NB15.csv /user/cloudera/hive\_project**

**Step 4:** The command "create database hive\_project" is used in Hive to create a new database named "hive\_project".

Assuming you have the necessary permissions to create a database in Hive, running this command will create a new database named "hive\_project" in Hive's metastore, which is a centralized repository for metadata about data stored in Hive.

Once the database is created, you can use it to store and organize tables, views, and other database objects in Hive. For example, you can create tables in the "hive\_project" database to store data in HDFS or other data storage systems, and then use Hive's SQL-like language to query and analyze the data.

**Step 5:**  The command you provided is used to create an external table named "unsw\_nb15" in the "hive\_project" database in Hive. The table schema includes a list of columns and their data types, as well as the location of the data in HDFS.

Assuming you have the necessary permissions to create tables in Hive and read data from the "/user/cloudera/hive\_project" directory in HDFS, running this command will create a new external table named "unsw\_nb15" in the "hive\_project" database, with the specified columns and data types. The table will be stored in the Hive metastore, but the data itself will be stored externally in HDFS at the location specified in the "LOCATION" clause.

The table is defined to use a comma-separated values (CSV) format, with fields terminated by a comma. The data file(s) should be stored in text format on HDFS.

This table is ready to be queried in Hive using SQL-like syntax, and can be used for data analysis and processing.



**Step 6:** In this step we have queried the table by **select \* from** command to show the data stored in the table **unsw\_nb15.**

Graphical user interface, application

Description automatically generated

* **Databricks:**

**Here are the steps with screen shots of queries and analysis using Databricks:**

**Step 1:** Login to your free Databricks Community Version Account

A screenshot of a login box

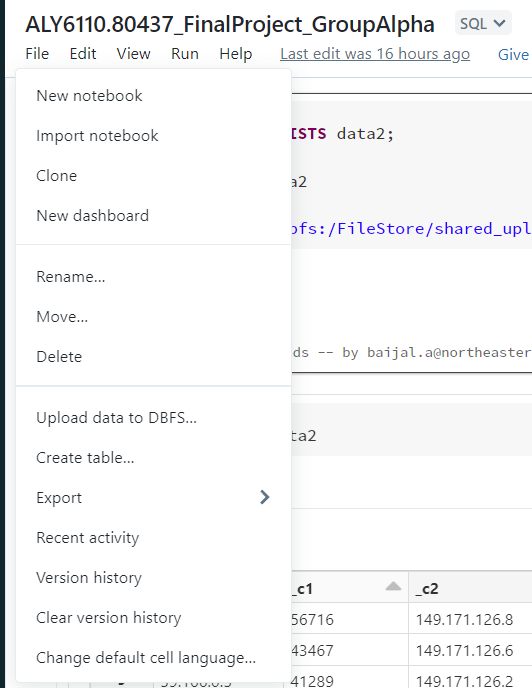
Description automatically generated with medium confidence

**Step 2:** Create a new Notebook by clicking on the first link

A screenshot of a computer

Description automatically generated with medium confidence

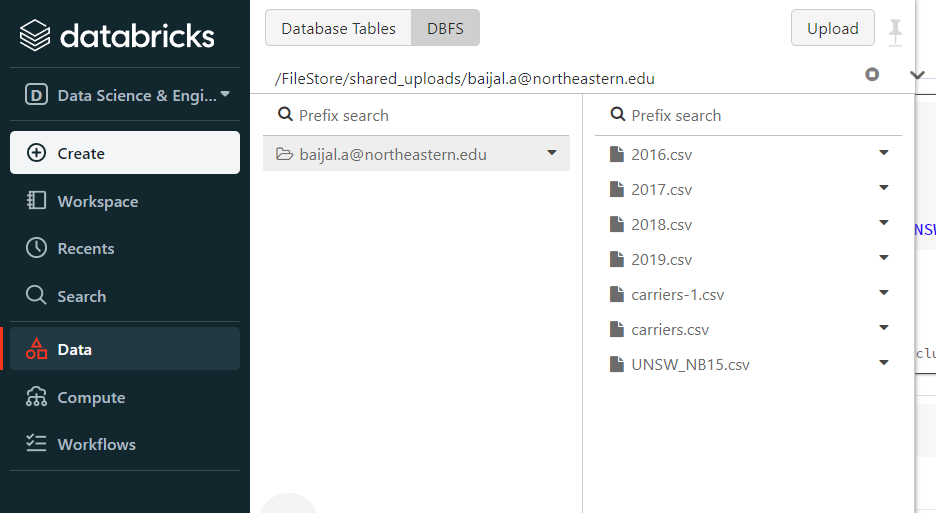
**Step 3:** Click on File > Upload Data to DBFS. Then the below screen will pop up where you can upload your dataset



A screenshot of a computer

Description automatically generated with medium confidence

Further, to check if your data has been uploaded or not, click on “Data” from the side menu > select DBFS > Select your account. Your dataset should appear among the files



**Step 4:** Next, to be able to run the cells containing the commands, we need a cluster. We “Create, Attach & Run” a cluster.

A screenshot of a computer

Description automatically generated

**Step 5:** Next, we select our default language. In this project, we chose SQL as the main default language.

A picture containing text, font, screenshot, white

Description automatically generated

**Step 6:** Now, the environment is ready to type in and execute various commands. First, we copy our data from the DBFS into a table “data2”. To avoid any duplicate data or overwriting, we drop any pre-existing table with the same name. Then, we create a new table which takes data from the dataset that was imported into DBFS.

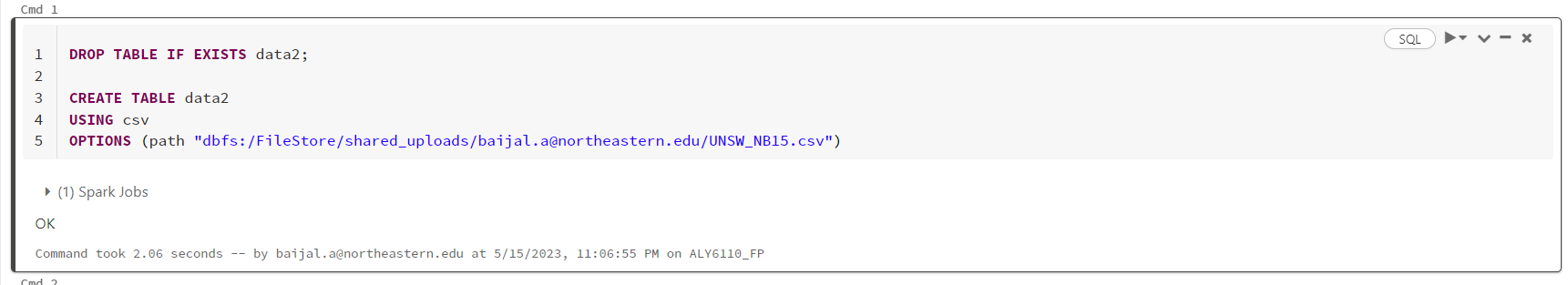
**DROP TABLE IF EXISTS data2;**

**CREATE TABLE data2**

**USING csv**

**OPTIONS(path “dbfs:/FileStore/shared\_uploads/baijal.a@northeastern.edu/UNSW\_NB15.csv”)**

The above command is also prompted to the user when the dataset was imported into DBFS.



**Step 7**: Finally, we check if the data was added to the table

**Select \* from data2**

A screenshot of a computer

Description automatically generated with medium confidence

**Analysis Outcomes and Insight Visualization:**

* **Hive:**

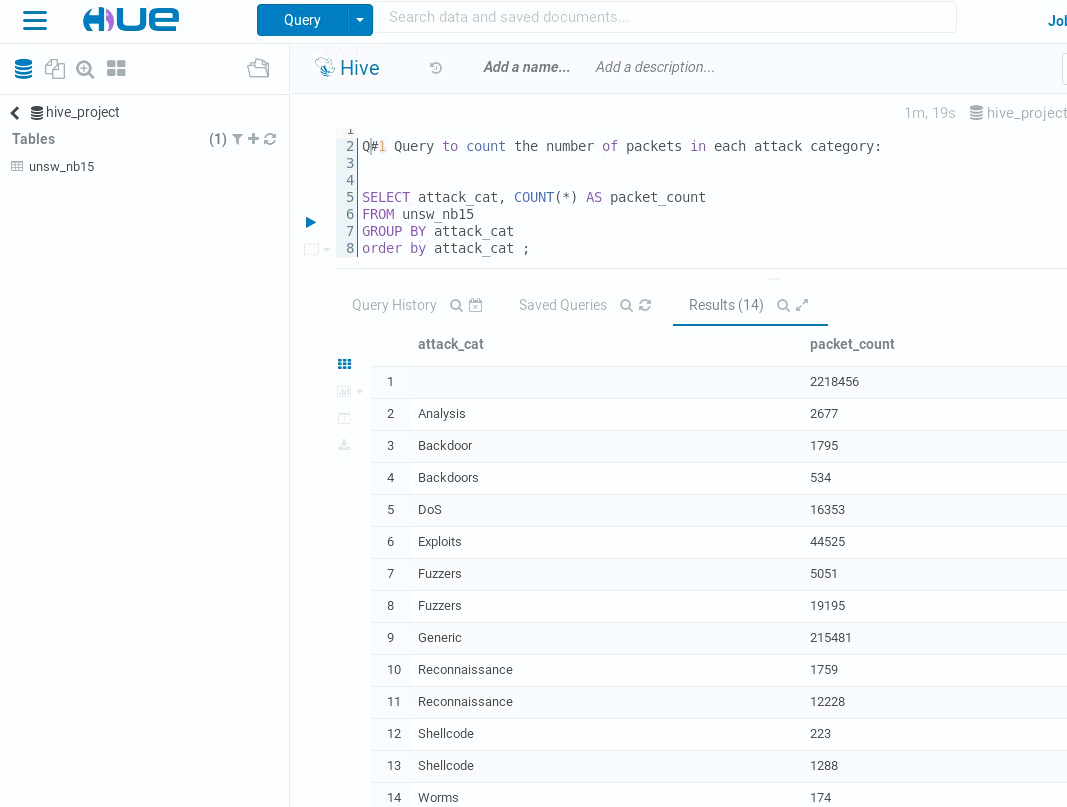
**1st Insight Query:** Count the number of packets in each attack category

**SELECT attack\_cat, COUNT(\*) AS packet\_count FROM unsw\_nb15 GROUP BY attack\_cat order by attack\_cat ;**

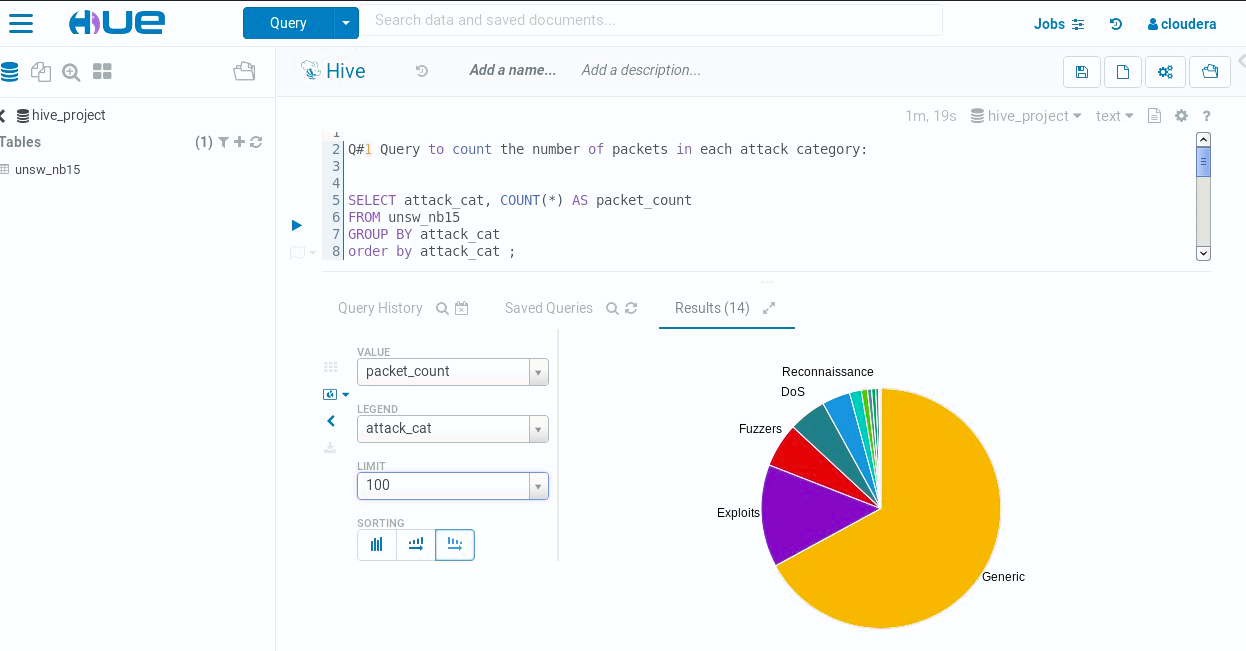
"unsw\_nb15" table containing network traffic data, the query selects two columns: "attack\_cat" and a count of the number of packets in each attack category ("packet\_count").

The "GROUP BY" clause groups the data by the "attack\_cat" column, so that the query returns the packet count for each distinct attack category. The "ORDER BY" clause orders the results by the attack category name in ascending order.

So, the output of this query will be a table with two columns: "attack\_cat" and "packet\_count", where each row represents an attack category and the number of packets in that category.



**Visualization:** to count the number of packets in each attack category using pie chart.



**2nd Insight Query:**  **To find the top 10 source IP addresses with the highest number of packets.**

**SELECT srcip, COUNT(\*) AS connection\_count**

**FROM unsw\_nb15**

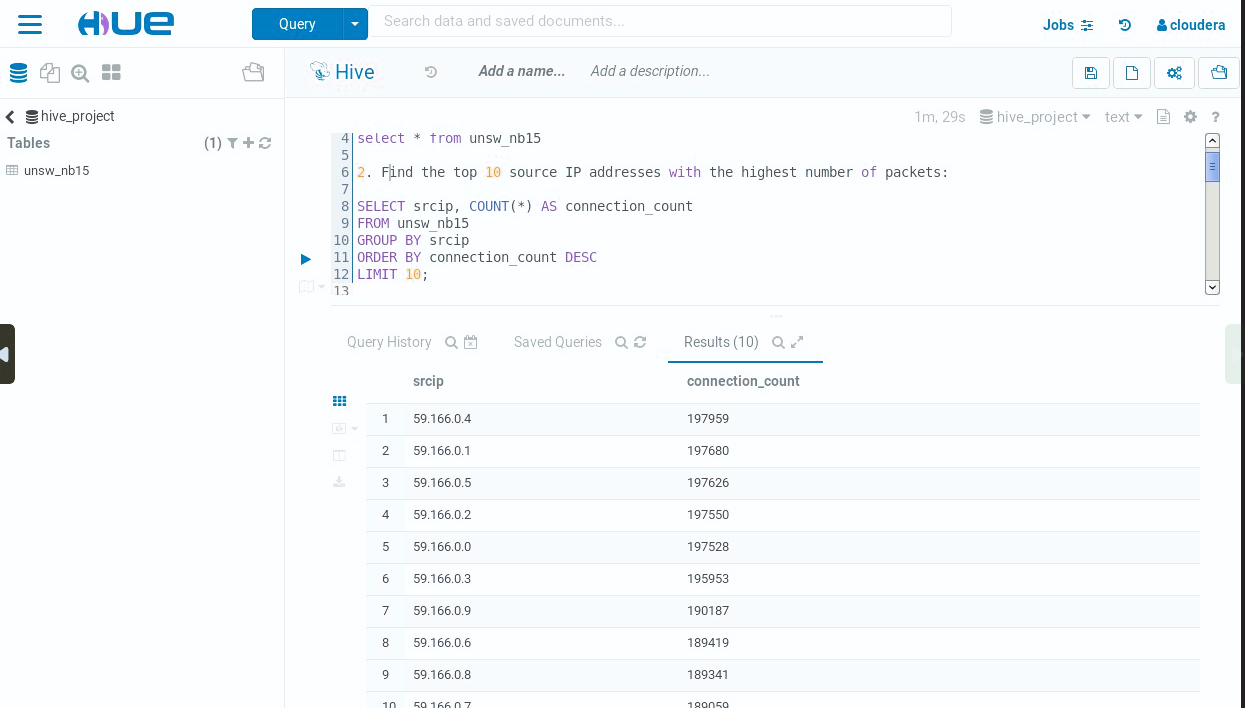
**GROUP BY srcip**

**ORDER BY connection\_count DESC**

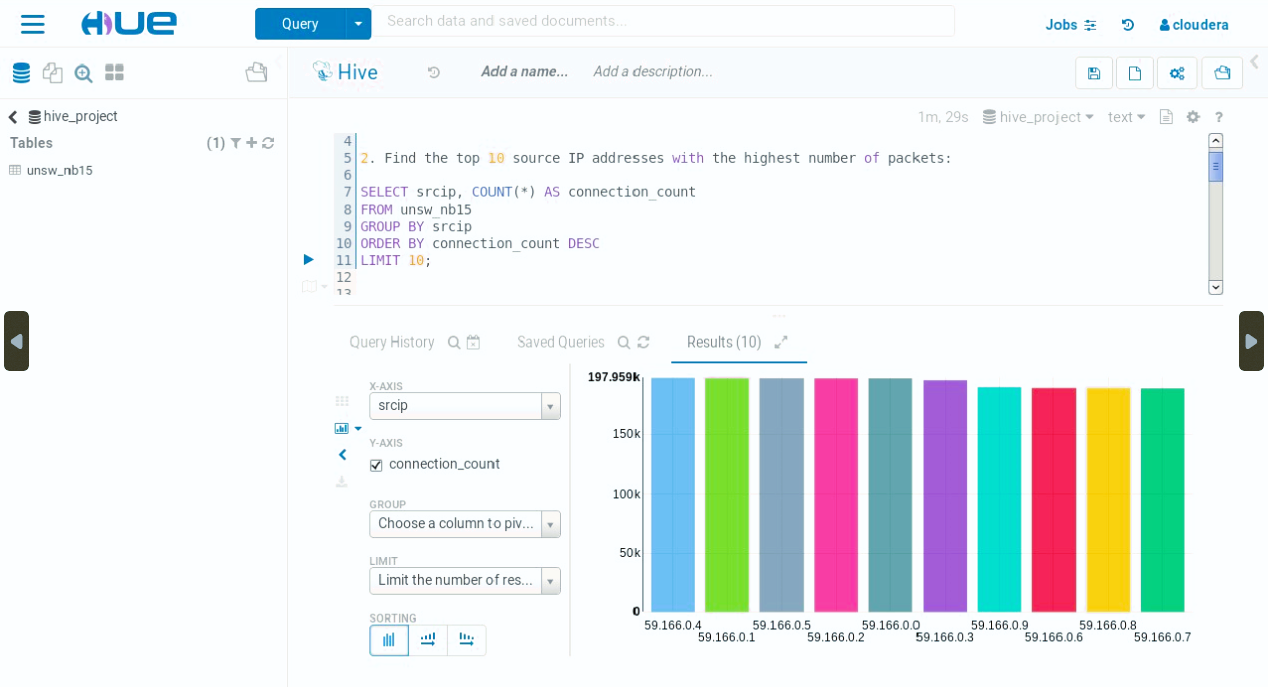
**LIMIT 10;**

This query can be useful for identifying potential sources of network traffic or attacks on a network. By analyzing the number of connections from each IP address, it can help detect any abnormal or suspicious behavior. It can also be used for network capacity planning, to identify which IP addresses are consuming the most bandwidth and resources.

It's worth noting that this query only provides a high-level overview of the network connections and does not provide any details about the specific type of traffic or attacks. To obtain more detailed information, additional queries or analysis may be necessary.



**Visualization:** This step Bar Graph is used to present the results of above 2nd query.

****

**3rd Insight Query:** Write a query to show the most common destination port numbers for each attack category.

**SELECT srcip, COUNT(\*) AS packet\_count**

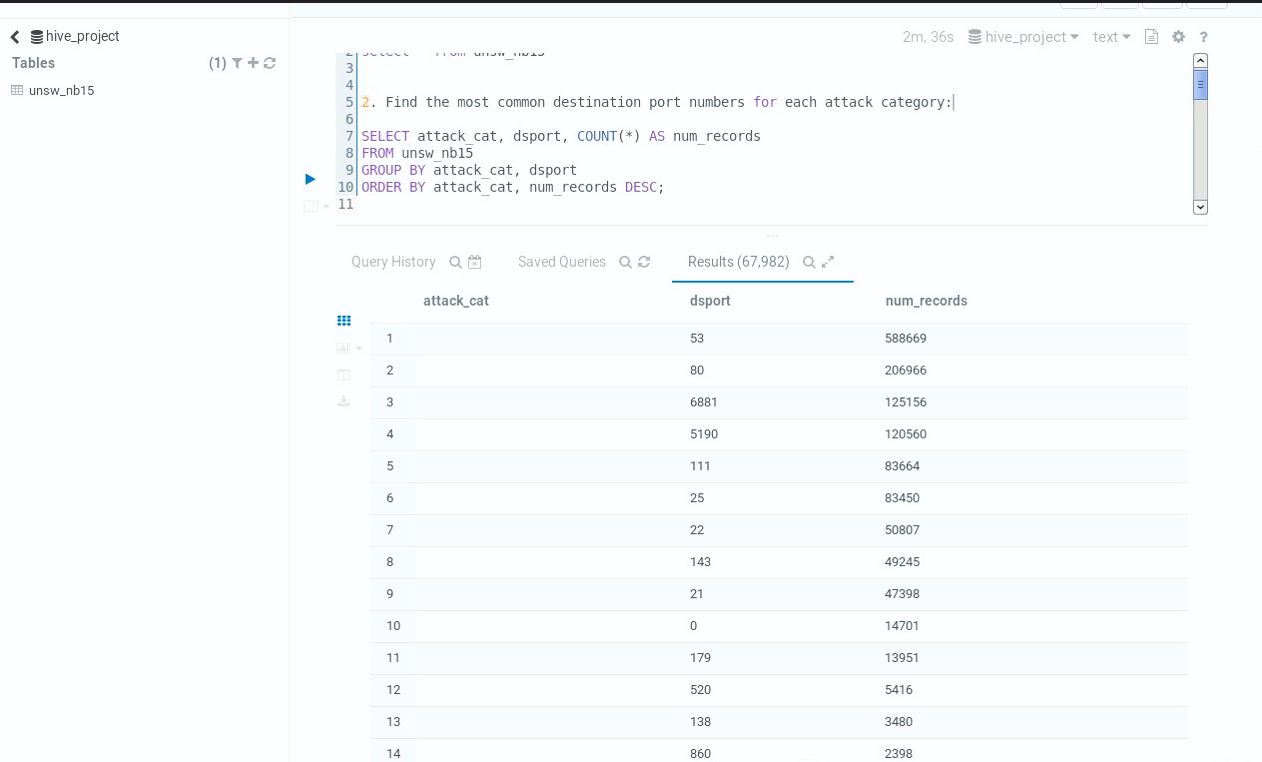
**FROM unsw\_nb15**

**GROUP BY srcip**

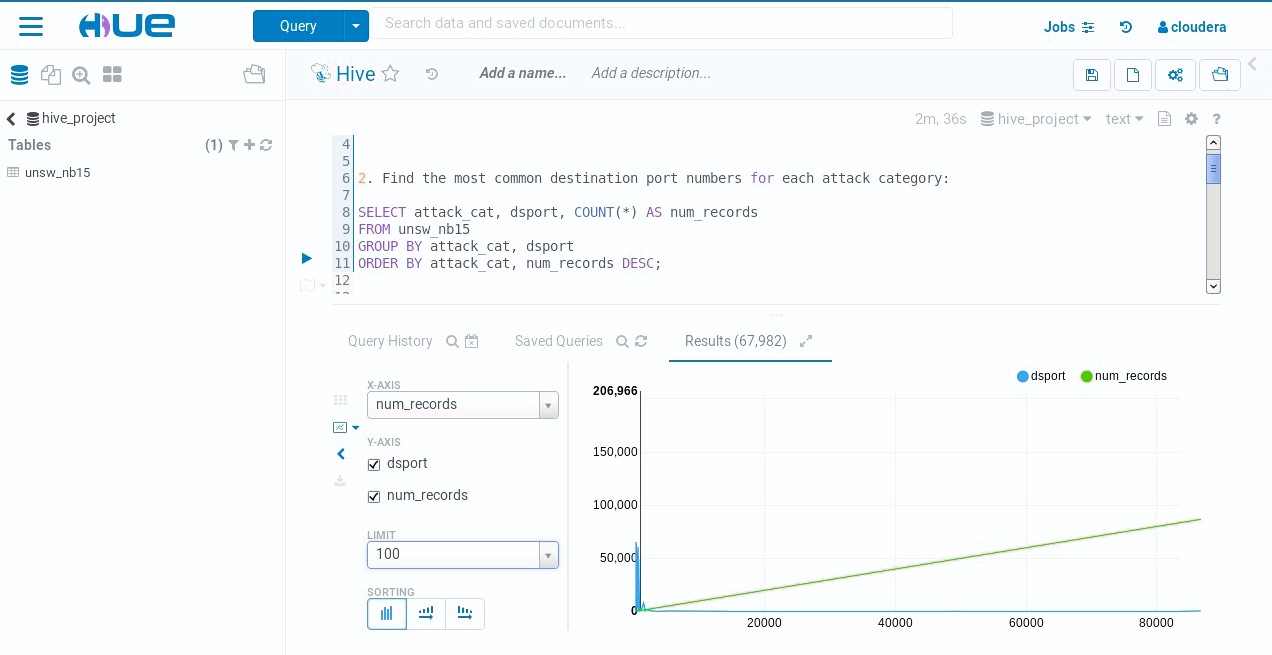
**ORDER BY packet\_count DESC**

**LIMIT 10;**

This query is useful for identifying which IP addresses are responsible for sending the most network traffic, which can be useful for identifying potential bandwidth bottlenecks or network security threats. However, it's important to note that this query alone doesn't provide any information about the type or quality of the traffic, and additional analysis may be required to fully understand the network activity.



**Visualization:** This step shows the graphical view of the above executed 3rd hive query.

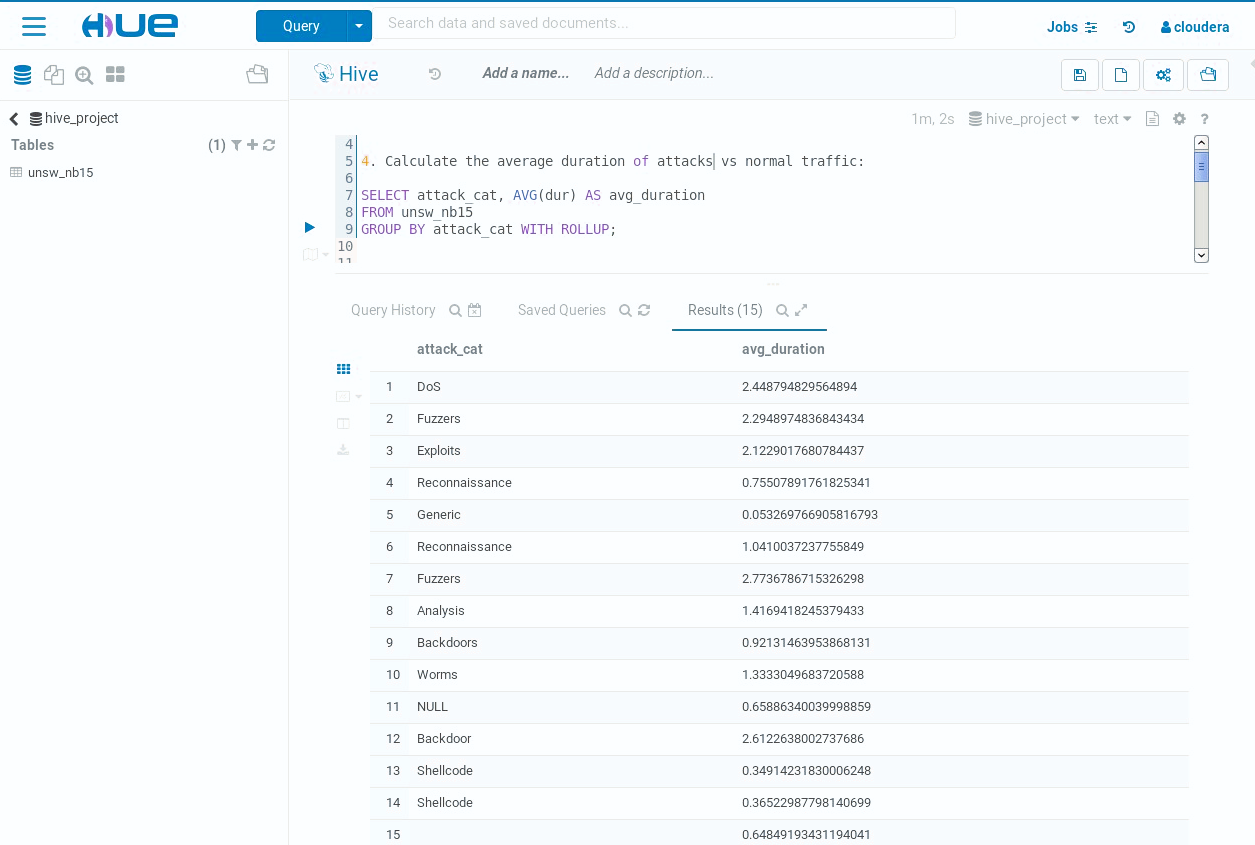


**4th Insight Query:** This step query is executed to show the average duration of attacks vs normal traffic.

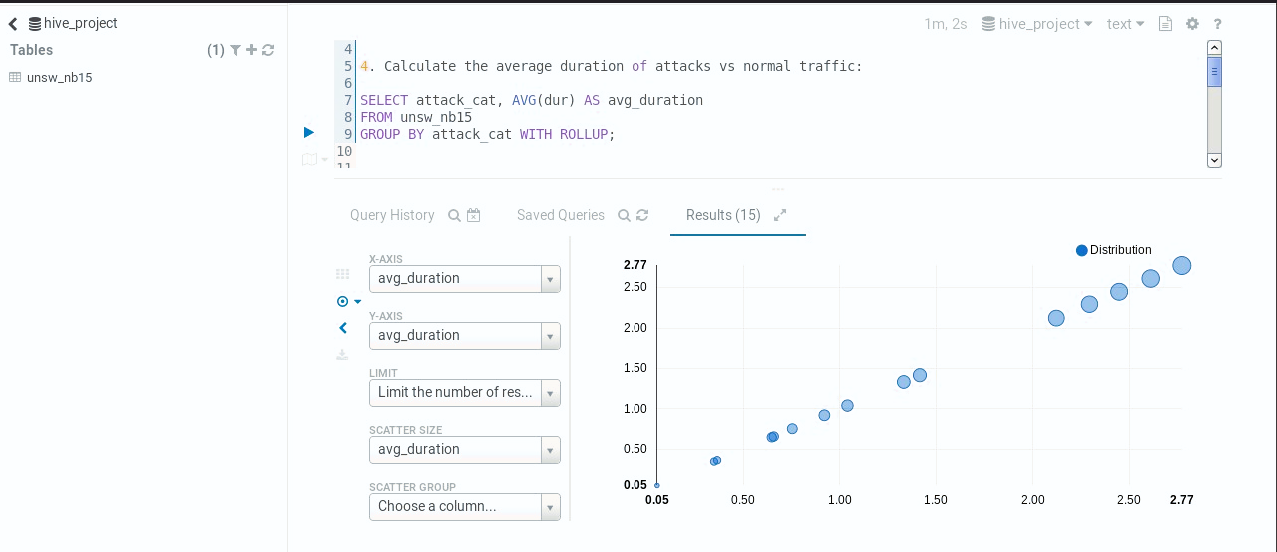
**SELECT attack\_cat, AVG(dur) AS avg\_duration**

**FROM unsw\_nb15**

**GROUP BY attack\_cat WITH ROLLUP;**

  
It's worth noting that this query only provides a high-level overview of the average duration for each attack category and does not provide any details about the specific attacks or their characteristics. Additionally, using the AVG function to calculate the average duration may not be the most informative approach.

**Visualization:** This step shows the graphical view of the above 4th query.



**5th Insight Query:** find the top 5 attacked services along with number of attacks.

**SELECT service, COUNT(\*) AS num\_attacks**

**FROM unsw\_nb15**

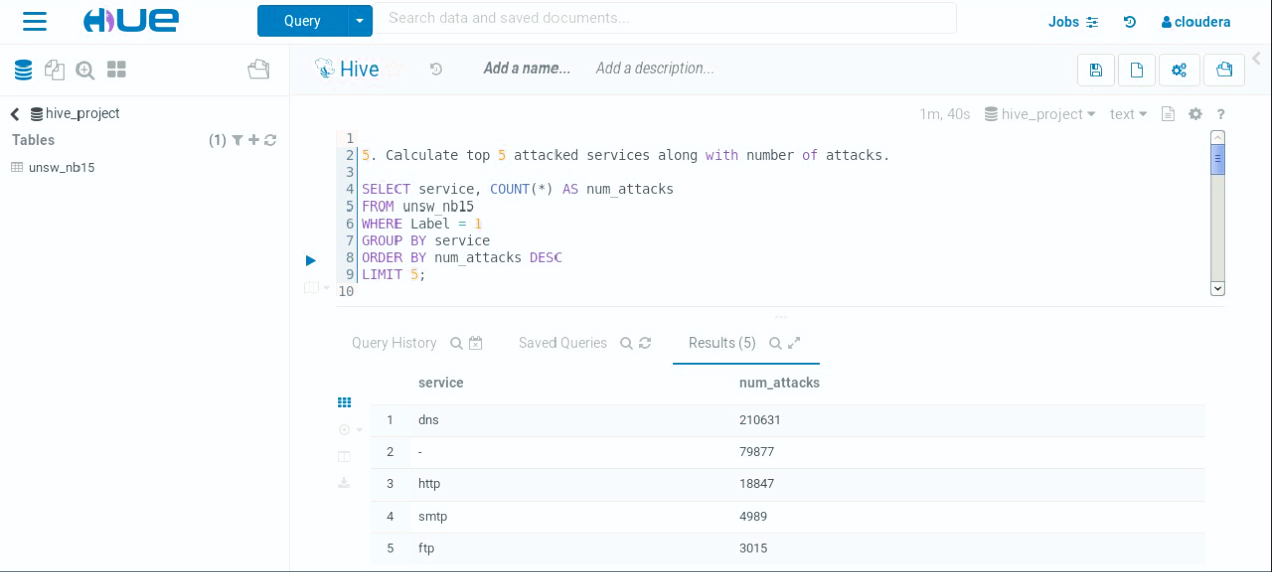
**WHERE Label = 1**

**GROUP BY service**

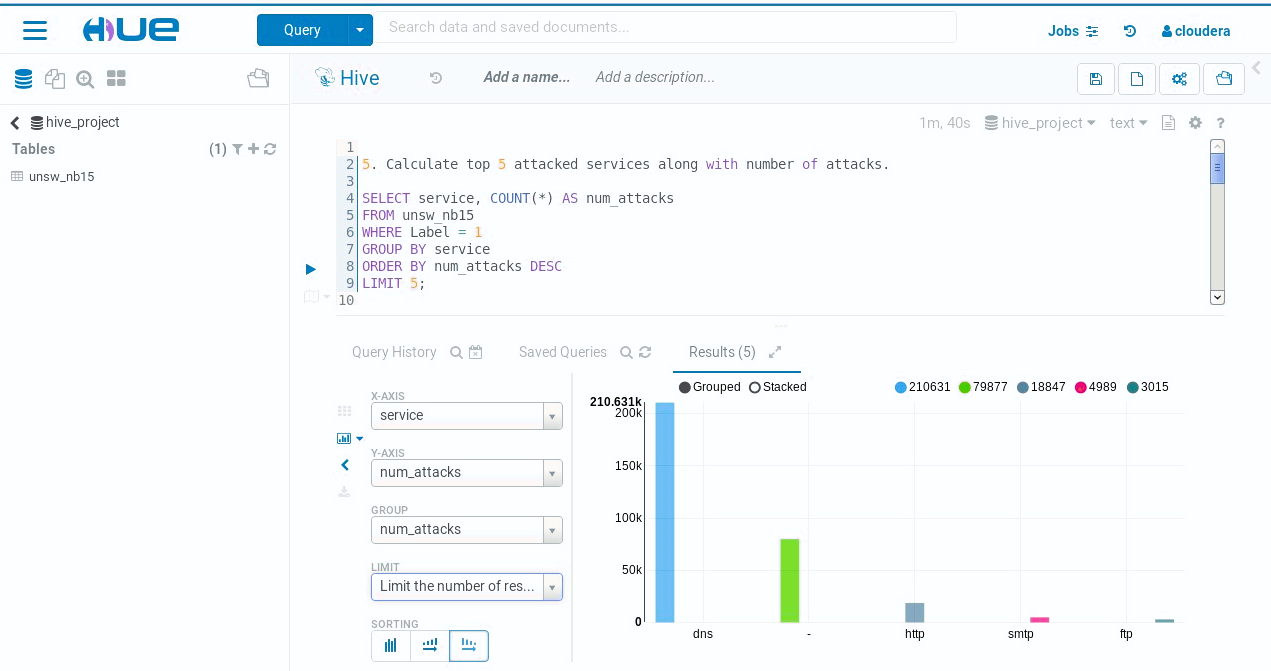
**ORDER BY num\_attacks DESC**

**LIMIT 5;**

This query can be useful for identifying which services are most targeted in attacks, as it provides a count of the number of attacks per service. By limiting the results to the top 5 services with the highest number of attacks, we can quickly see which services are the most popular targets. However, this query doesn't provide any information about the duration or severity of the attacks on each service, so further analysis may be required to fully understand the nature of the attacks.



**Visualization:** This step shows the graphical view of the above query.



* **Databricks:**

**1st Insight Query:** Type of Label on network attacks

**select \_c48 as label, count(\_c48) as label\_count from data2 group by \_c48**

“data2” is the table that contains the data from the file “UNSW\_NB15”. The alias **“\_c48**” column represents the binary classification of type of label attack. We use “**label\_count**” to count the total values of each “**label**”. The labels of 1 and 0 are often used to indicate the classification or categorization of network traffic or network events as either an attack or non-attack (normal) activity. ​This label 0 is assigned to network traffic or events that are considered normal or legitimate activities within the network​. The label 1 is assigned to network traffic or events that are identified as malicious or unauthorized activities. By assigning these labels to network data, network security systems and machine learning algorithms can be trained to distinguish between malicious and non-malicious attacks. We can clearly observer here that almost 2.25M of the data we have represent a non-attack label.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing rectangle, screenshot, electric blue, blue

Description automatically generated

**2nd Insight Query:** Number of different type of data transfer rates of attacks

**select \_c9 as transfer\_rate, count(\_c9) as transfer\_rate\_count from data2 group by \_c9**

Columns containing the rate of the attacks, using an alias “**transfer\_rate**”, are extracted along with the count of each of these attacks, “**transfer\_rate\_count**”. Grouping is performed using the “group by” function. In network security, Transfer rate refers to the speed or rate at which data is transmitted or transferred over a network or communication channel. It is a measure of the amount of data that can be transmitted per unit of time.​

It is typically measured in bits per second (bps) or its multiples​ such as kilo bytes(kbps) and megabytes (mbps). As we can observe here most of the transfer rate is '31' at once in a particular period.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing screenshot, line, diagram, rectangle

Description automatically generated

**3rd Insight Query:** Different type of network packets on the networks.

**select \_c5 as type\_of\_network\_packet, count(\_c5) as count\_network\_packets from data2 group by \_c5**

Columns containing the type of network packets, use the alias “**type\_of\_network\_packet**”, are extracted along with the count of each of these packets, “**count\_network\_packets**”. Grouping then is performed using the “group by” function. In the context of network assaults, the initials "INT," "FIN," and "CON" denote different sorts of network packets or flags used in network communication protocols, most notably the TCP/IP suite of protocols. These packet flags (INT/SYN, FIN, and CON/ACK) are essential for establishing and terminating TCP connections and ensuring reliable data transmission across networks. "FIN" which stands for "Finish" is notably the highest type of network packet, which is obvious and can be seen in data with over 1.5M count.

A screenshot of a computer

Description automatically generated

* **Visualization:**

A picture containing text, plot, screenshot, line

Description automatically generated

**4th Insight Query:** Number of different attacks on applications with OSPF protocol

**select \_c47 as Attack\_Category, count(\_c47) as Attack\_Category\_Count from data2 where \_c4=="ospf" group by \_c47**

“data2” is the table that contains the data from the file “UNSW\_NB15”. Columns containing the name of the attacks, using an alias “Attack\_Category”, are extracted along with the count of each of these attacks, “Attack\_Category\_Count”. These are again filtered by selecting rows that have application protocol (\_c4) as “ospf”. Finally, grouping is performed using the “group by” function.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing text, plot, line, diagram

Description automatically generated

As can be observed from the result, the category “Exploits” was the most frequent type of attack, followed by “DOS”. Also, the least common attacks are “Reconnaissance” and “Backdoors” attacks.

**5th Insight Query:** Number of different attacks on applications with TCP protocol

**select \_c47 as Attack\_Category, count(\_c47) as Attack\_Category\_Count from data2 where \_c4=="tcp" group by \_c47**

Similarly, using the alias “Attack\_Category”, columns providing the names of the attacks are extracted, coupled with columns giving the number of each attack, “Attack\_Category\_Count”. Once more, they are filtered by choosing rows with “tcp” as the application protocol (\_c4). The “group by” function is then used to group the data.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing text, diagram, font, line

Description automatically generated

Once more, it was discovered that "Exploits" were the most typical assault against TCP-enabled programs. Attacks known as "Fuzzers”, and "Reconnaissance" come next.

**6th Insight Query:** Number of different attacks on applications with UDP protocol

**select \_c47 as Attack\_Category, count(\_c47) as Attack\_Category\_Count from data2 where \_c4=="udp" group by \_c47**

Likewise, using the alias “Attack\_Category”, columns containing the names of the assaults are retrieved together with columns containing the total number of attacks, “Attack\_Category\_Count”. By selecting rows with “udp” as the application protocol (\_c4), they are once more filtered. After that, the data is grouped using the “group by” function.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing line, screenshot, plot, diagram

Description automatically generated

For UDP, it is very different from the other two protocols that were explored. For applications utilizing UDP, “Generic” attacks were the most common. Followed by “Fuzzers” and “Reconnaissance” attacks. However, there is a big difference between number of times these attacks have occurred when compared to the “Generic” type.

One of the most common network attacks is Denial of Service Attacks, also known as “DOS” Attacks. These attacks are malicious attempts to prevent a computer system, network, or online service from operating normally and rendering them unavailable to authorized users. A DOS attack aims to deprive legitimate users of service by overloading the target's resources, such as its network bandwidth, computing power, or memory.

**7th Insight Query:** What percentage of attacks are there against each kind of network protocol?

**select \_c4 as Network\_Protocol, avg(\_c6) as Average\_Attack\_Time from data2 where \_c47=="DoS" group by \_c4 order by Average\_Attack\_Time desc limit 10**

First, extract the column containing the network protocols, using an alias “Network\_Protocol” and also calculate average time of attack for these different network protocols using the function “avg()” with an alias “Average\_Attack\_Time”. These are further filtered for only systems that were faced with a “DOS” attack. Since the list is very long, only the top 10 results are printed.

A screenshot of a computer

Description automatically generated with medium confidence

* **Visualization:**

A picture containing screenshot, diagram, plot

Description automatically generated

From the above output, it can be observed that, on average, systems with OSPF, Open Shortest Path First Protocol have the longest DOS attack time, followed by systems with RVD (Rendezvous daemon), SCTP (Stream Control Transmission Protocol), TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) configurations. Also, a major difference can be seen between the attack times for OSPF, RVD AND SCTP. However, the difference between the attack times for SCTP, TCP and UDP is not much. This means that more research needs to be done for the OSPF protocol as it has the highest attack time, making the system vulnerable for the longest time.

**Project Summary:**

Network attacks are malicious acts meant to jeopardize the resourcefulness, availability, or secrecy of computer networks. These assaults may have detrimental effects, ranging from loss of network services to illegal access to sensitive data.

To find trends and insights on network security, this study used the UNSW-NB15 dataset, a sizable dataset collecting statistics on network traffic. The project required processing and analyzing the dataset with Hadoop and Cloudera as well as employing Hive queries and visualizations to glean insights and respond to certain network security-related queries. Databricks was also utilized in this project.

The most important project steps involved importing and preparing the dataset, creating a Hive table to store the data, and several insightful queries, including counting the number of packets in each attack category, finding the top source IP addresses, identifying the most popular destination port numbers for each attack category, comparing the average attack duration to normal traffic, and figuring out the top attacked sites. Identification of prospective network security concerns, comprehension of typical attack patterns, and identification of potential areas for network capacity planning and optimization were among the project's anticipated outcomes.

The project's overall goal was to emphasize the value of network security in today's technology-driven society and to show how big data analytics and visualization tools may be used to derive useful insights from large-scale datasets of network traffic.

**Comments:**

UNSW-NB15 (University of New South Wales-Network-Based 15) is a freely available dataset that is widely utilized in network security research. The Australian Centre for Cyber Security (ACCS) at the University of New South Wales created it.

Here are some reactions to UNSW-NB15:

The UNSW-NB15 dataset was produced with the goal of providing a comprehensive and realistic dataset for testing and improving network intrusion detection systems (NIDS). It contains a diverse spectrum of network traffic data, including both normal and malicious actions, making it an invaluable resource for network security research.

Size and Variety: With over 2.5 million records, the dataset is excellent for training and testing machine learning models. It includes a wide range of attack types, such as DoS, DDoS, port scanning, and botnet attacks. This variety allows researchers to investigate various intrusion detection techniques.

Realistic Traffic: One of UNSW-NB15's strengths is its realistic simulation of network traffic. It was created by fusing real network data acquired in a controlled environment with fake traffic. This combination produces a more accurate representation of real-world network behavior, increasing the dataset's utility.

Feature Selection: The dataset includes 49 features, including protocol-related information, source and destination IP addresses, port numbers, packet length, and time-related information. These features allow researchers to extract meaningful insights and develop effective intrusion detection models.

UNSW-NB15, like any other dataset, has some limitations. One of the most common objections is that the dataset does not include encrypted communication, which is a critical component of modern network security. Furthermore, because the dataset is old (it was released in 2015), it may not contain the most recent attack strategies and trends.

Despite its limitations, the UNSW-NB15 has been extensively adopted and has contributed significantly to advances in intrusion detection research. It has been utilized in various research to build and assess machine learning algorithms, deep learning models, and anomaly detection techniques, promoting innovation in network security.

Overall, the project had **several pros and cons**. One of the main pros was the availability of the UNSW-NB15 dataset, which provided a large and diverse dataset for analysis. Using Cloudera and Hadoop for processing and Hive for querying, it was relatively easy to execute the tasks and gain insights into the data. The queries executed provided useful insights, such as the number of packets in each attack category and the top source IP addresses with the highest number of packets.

However, there were some **difficulties and issues** encountered during the project. One of the main challenges was the need for expertise in Cloudera, Hadoop, and Hive to effectively execute the queries and gain meaningful insights. The analysis required a strong understanding of big data concepts and techniques, which may be a barrier to entry for some users.

Additionally, while the insights gained were useful, they were high-level and lacked specific details about the attacks and their characteristics. This limited the ability to fully understand the nature of the attacks and the network traffic. Overall, the project provided valuable experience in working with big data and demonstrated the potential for using Cloudera, Hadoop, and Hive for analyzing network traffic data.

**Conclusion:**

Finally, the UNSW-NB15 dataset is a great resource for network security research, particularly for evaluating and developing intrusion detection algorithms. Its huge scale, wide range of attack methods, and accurate modeling of network traffic are all advantages. The extensive set of features in the dataset enables useful insights and the construction of effective intrusion detection models.

Despite these drawbacks, the UNSW-NB15 dataset has had a considerable impact on network security research. It has been widely implemented and has aided developments in intrusion detection techniques, such as the creation and evaluation of machine learning algorithms and anomaly detection methodologies.

Working with the UNSW-NB15 dataset using Cloudera, Hadoop, and Hive was a valuable learning experience. The dataset provided a rich source of data for exploring network traffic and identifying potential security threats. The use of Hive queries allowed for efficient analysis and visualization of the data, which helped in gaining insights and identifying patterns.

The UNSW-NB15 dataset is an important resource for network security and intrusion detection researchers and practitioners, and it continues to contribute to improvements in the field.

One of the main challenges was dealing with the large size of the dataset, which required careful management of resources and optimizing queries. However, this also provided an opportunity to learn about the importance of data processing and management in big data projects.

Overall, the project provided a good understanding of the capabilities of Hadoop and Cloudera for big data analysis and their potential applications in the field of cybersecurity. Moving forward, there is potential for further exploration of the dataset using more advanced machine learning techniques to improve the accuracy of identifying and preventing network attacks.

In conclusion, the project was a valuable learning experience and provided a good foundation for future work in big data analysis and cybersecurity.

**References:**

[1] Moustafa, Nour, and Jill Slay. "*UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set).*" Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[2] T. White (2012), “*Hadoop: The definitive guide*:” O’Reilly Media, Inc.

[3] Moustafa, Nour, and Jill Slay. "*The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset.*" Information Security Journal: A Global Perspective (2016): 1-14.

[4] Moustafa, Nour, et al. “*Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks.”* IEEE Transactions on Big Data (2017).

[5] Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. “*NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems”*. In Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings (p. 117). Springer Nature.

[6] UNSW-NB15 Available <https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15>