Tobi Ogunmokun

Technical Skills Portfolio

Python, SQL, Linux, and Incident Reporting Projects

CONTACT

Tobiomahal@gmail.com

(832)-305-3130

Linux Portfolio

Linux Command Line and Scripting Fundamentals

This Mini-project showcases my foundational Linux knowledge, including command-line navigation, file management, permissions handling, and basic scripting. I have used these skills on multiple personal projects including My Homelab DNS Server.

I will be demonstrating my skills on Bash through windows as i am unable to access my Pi5

File System Navigation and Management

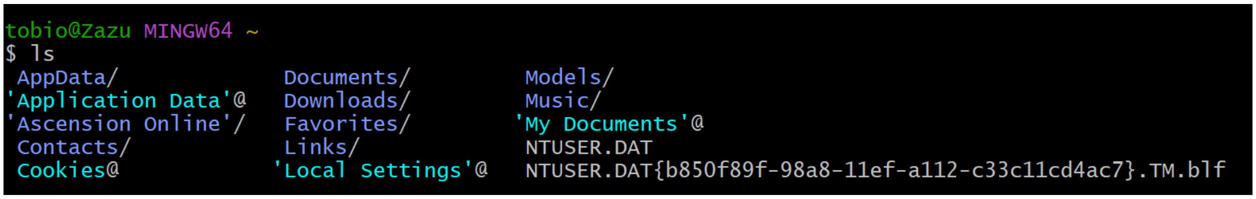
Commands Used:

cd /home/user/documents
ls -lah
mkdir playground
touch something.txt
mv report.txt ProjectFiles/renamed_file.txt

OBJECTIVE

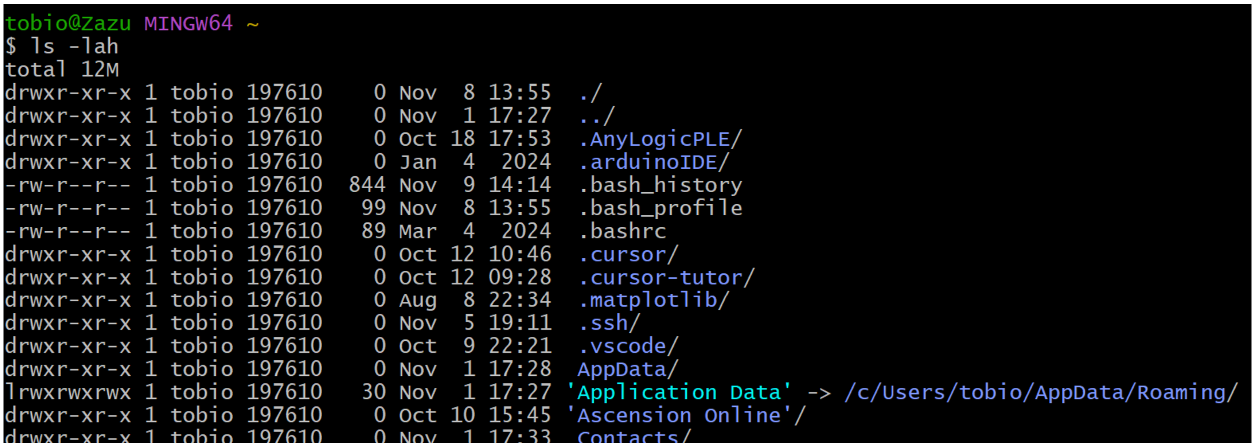
Showcased directory navigation, file listing, directory creation, and file movement.

`ls` to check what's in the directory i am currently on



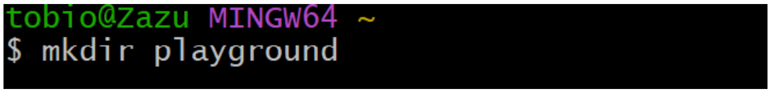
```
tobio@Zazu MINGW64 ~
$ ls
 AppData/              Documents/          Models/
'Application Data'@    Downloads/          Music/
'Ascension Online'/    Favorites/         'My Documents'@
 Contacts/             Links/              NTUSER.DAT
 Cookies@             'Local Settings'@    NTUSER.DAT{b850f89f-98a8-11ef-a112-c33c11cd4ac7}.TM.blf
```

`Ls -lah` also lists the files in the directory but it shows a lot more detail like who has rwx permissions as well as direct directories.

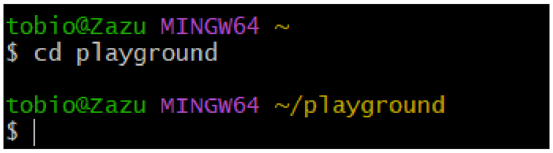


```
tobio@Zazu MINGW64 ~
$ ls -lah
total 12M
drwxr-xr-x 1 tobio 197610    0 Nov  8 13:55  ./
drwxr-xr-x 1 tobio 197610    0 Nov  1 17:27  ../
drwxr-xr-x 1 tobio 197610    0 Oct 18 17:53  .AnyLogicPLE/
drwxr-xr-x 1 tobio 197610    0 Jan  4  2024  .arduinoIDE/
-rw-r--r-- 1 tobio 197610  844 Nov  9 14:14  .bash_history
-rw-r--r-- 1 tobio 197610   99 Nov  8 13:55  .bash_profile
-rw-r--r-- 1 tobio 197610   89 Mar  4  2024  .bashrc
drwxr-xr-x 1 tobio 197610    0 Oct 12 10:46  .cursor/
drwxr-xr-x 1 tobio 197610    0 Oct 12 09:28  .cursor-tutor/
drwxr-xr-x 1 tobio 197610    0 Aug  8 22:34  .matplotlib/
drwxr-xr-x 1 tobio 197610    0 Nov  5 19:11  .ssh/
drwxr-xr-x 1 tobio 197610    0 Oct  9 22:21  .vscode/
drwxr-xr-x 1 tobio 197610    0 Nov  1 17:28  AppData/
lrwxrwxrwx 1 tobio 197610   30 Nov  1 17:27 'Application Data' -> /c/Users/tobio/AppData/Roaming/
drwxr-xr-x 1 tobio 197610    0 Oct 10 15:45 'Ascension Online'/
drwxr-xr-x 1 tobio 197610    0 Nov  1 17:33  Contacts/
```

I will create a new directory to play around in using mkdir

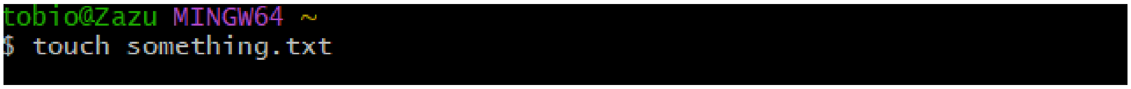


```
tobio@Zazu MINGW64 ~
$ mkdir playground
```

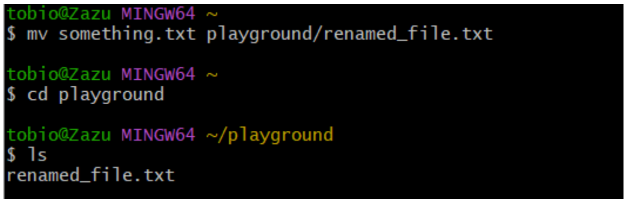


```
tobio@Zazu MINGW64 ~
$ cd playground

tobio@Zazu MINGW64 ~/playground
$ |
```

touch something.txt will create a .txt file called “something” in whatever directory i am currently working in.

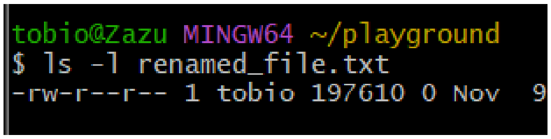


```
tobio@Zazu MINGW64 ~
$ touch something.txt
```



```
tobio@Zazu MINGW64 ~
$ mv something.txt playground/renamed_file.txt

tobio@Zazu MINGW64 ~
$ cd playground

tobio@Zazu MINGW64 ~/playground
$ ls
renamed_file.txt
```

I moved something.txt from my home dir. Using mv and into the playground dir. while renaming it to “renamed_file.txt”.



```
tobio@Zazu MINGW64 ~/playground
$ ls -l renamed_file.txt
-rw-r--r-- 1 tobio 197610 0 Nov  9
```

Checking permissions of my created .txt file with ls -l

I have read and write permissions and *group* and *others* can only read

Understanding and Modifying File Permissions

Commands Used:

cd playground

ls -l renamed_file.txt

echo '#!/bin/bash\necho "Hello, World!"' > example_script.sh

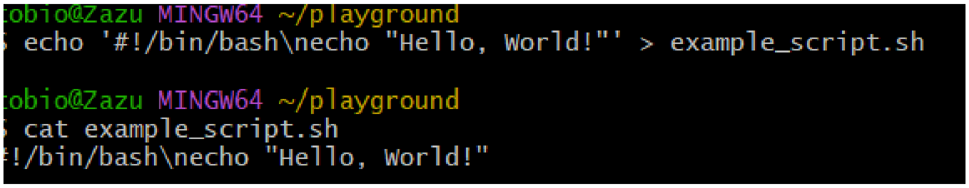
chmod +x example_script.sh

./example_script.sh

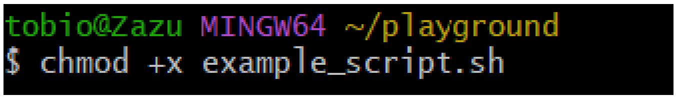
chmod 644 renamed_file.txt

OBJECTIVE

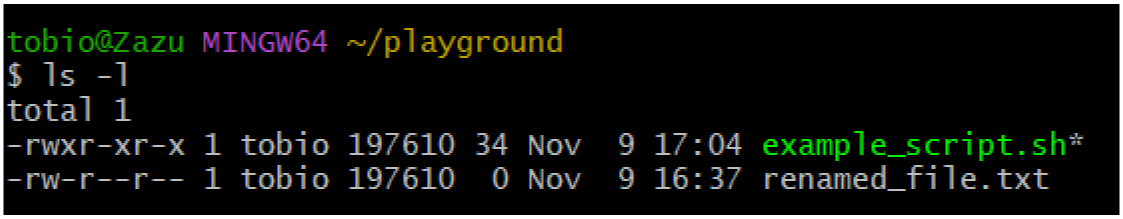
Demonstrated how to check and modify file permissions, create a simple script, make it executable, and adjust access levels.



```
obio@Zazu MINGW64 ~/playground
 echo '#!/bin/bash\necho "Hello, World!"' > example_script.sh

obio@Zazu MINGW64 ~/playground
 cat example_script.sh
!/bin/bash\necho "Hello, World!"
```



```
tobio@Zazu MINGW64 ~/playground
$ chmod +x example_script.sh
```



```
tobio@Zazu MINGW64 ~/playground
$ ls -l
total 1
-rwxr-xr-x 1 tobio 197610 34 Nov  9 17:04 example_script.sh*
-rw-r--r-- 1 tobio 197610  0 Nov  9 16:37 renamed_file.txt
```

Created an executable file called *example_script.sh*, then made it executable

Basic Text Processing

Commands Used:

head -n 10 renamed_file.txt

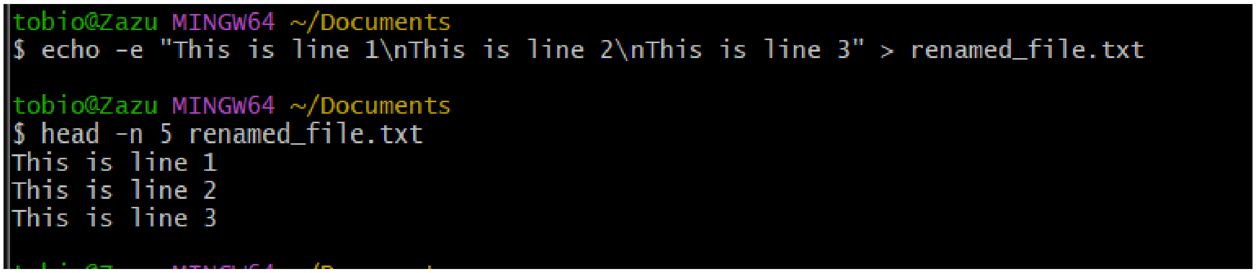
grep -i 'error' log.txt

sort data.txt | uniq

sort data.txt | uniq -c | sort -nr

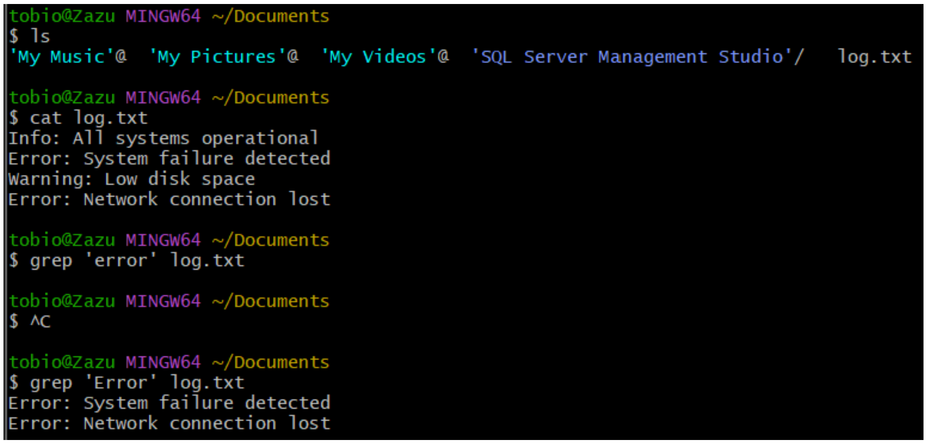
OBJECTIVES

Demonstrated file previewing, text search, sorting, and counting unique entries.



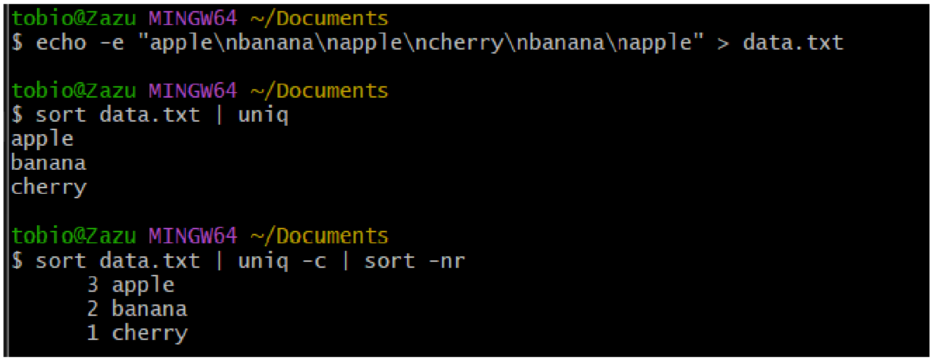
```
tobio@Zazu MINGW64 ~/Documents
$ echo -e "This is line 1\nThis is line 2\nThis is line 3" > renamed_file.txt

tobio@Zazu MINGW64 ~/Documents
$ head -n 5 renamed_file.txt
This is line 1
This is line 2
This is line 3
```

Added three lines of text to "renamed_file.txt" then read the first 5 lines in the file using head -n x



```
tobio@Zazu MINGW64 ~/Documents
$ ls
'My Music'@  'My Pictures'@  'My Videos'@  'SQL Server Management Studio'/   log.txt

tobio@Zazu MINGW64 ~/Documents
$ cat log.txt
Info: All systems operational
Error: System failure detected
Warning: Low disk space
Error: Network connection lost

tobio@Zazu MINGW64 ~/Documents
$ grep 'error' log.txt

tobio@Zazu MINGW64 ~/Documents
$ ^C

tobio@Zazu MINGW64 ~/Documents
$ grep 'Error' log.txt
Error: System failure detected
Error: Network connection lost
```

Created a log.txt file with an error message used cat to directly see the contents and also used grep to search for specific terms within the log.txt



```
tobio@Zazu MINGW64 ~/Documents
$ echo -e "apple\nbanana\napple\ncherry\nbanana\napple" > data.txt

tobio@Zazu MINGW64 ~/Documents
$ sort data.txt | uniq
apple
banana
cherry

tobio@Zazu MINGW64 ~/Documents
$ sort data.txt | uniq -c | sort -nr
      3 apple
      2 banana
      1 cherry
```

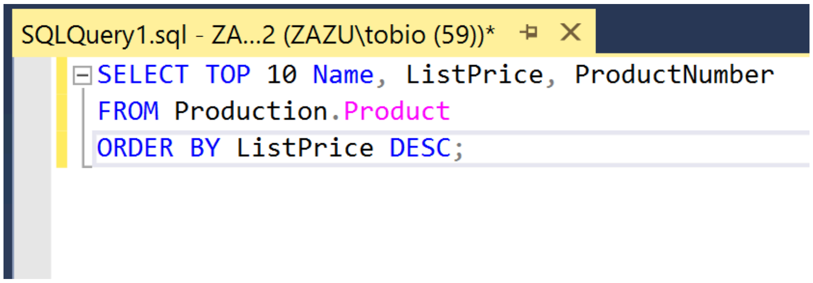
_Added sample data with duplicate lines to data.txt
_Sorted the contents of data.txt and removed duplicates
_Count occurrences of each unique line in data.txt

SQL Server Portfolio

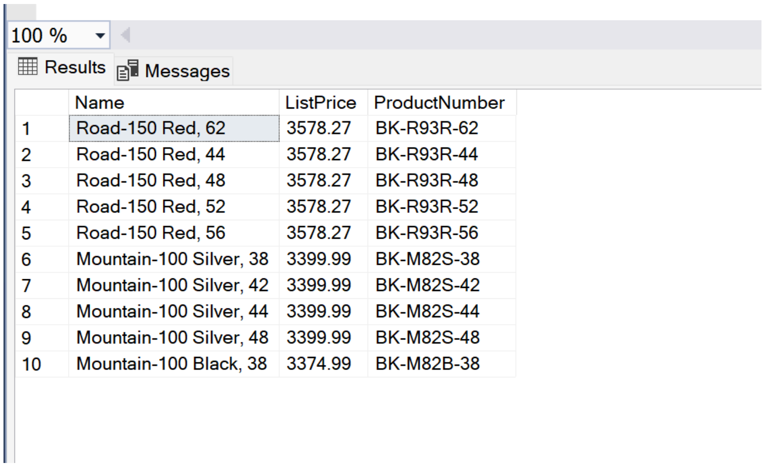
SQL Server Analysis Using Microsoft's Free AdventureWorks2022 Database.

<u>Exploring Product Data:</u>

First thing I did was take a look at the product data in the production schema/grouping. I ran a simple query to select the 10 most expensive products ordered by listing price from highest to lowest price.



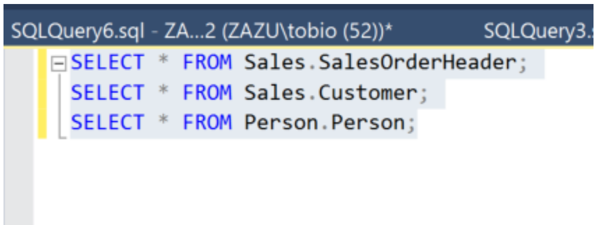
```sql
SELECT TOP 10 Name, ListPrice, ProductNumber
FROM Production.Product
ORDER BY ListPrice DESC;
```



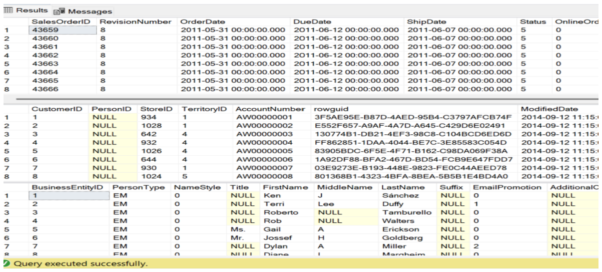
	Name	ListPrice	ProductNumber
1	Road-150 Red, 62	3578.27	BK-R93R-62
2	Road-150 Red, 44	3578.27	BK-R93R-44
3	Road-150 Red, 48	3578.27	BK-R93R-48
4	Road-150 Red, 52	3578.27	BK-R93R-52
5	Road-150 Red, 56	3578.27	BK-R93R-56
6	Mountain-100 Silver, 38	3399.99	BK-M82S-38
7	Mountain-100 Silver, 42	3399.99	BK-M82S-42
8	Mountain-100 Silver, 44	3399.99	BK-M82S-44
9	Mountain-100 Silver, 48	3399.99	BK-M82S-48
10	Mountain-100 Black, 38	3374.99	BK-M82B-38

<u>Customer Sales Analysis</u>

Then i wanted to examine the data in *SaleOrderHeader*, *Customer* and *Person* Databases to see what i can pull from them



```sql
SELECT * FROM Sales.SalesOrderHeader;
SELECT * FROM Sales.Customer;
SELECT * FROM Person.Person;
```



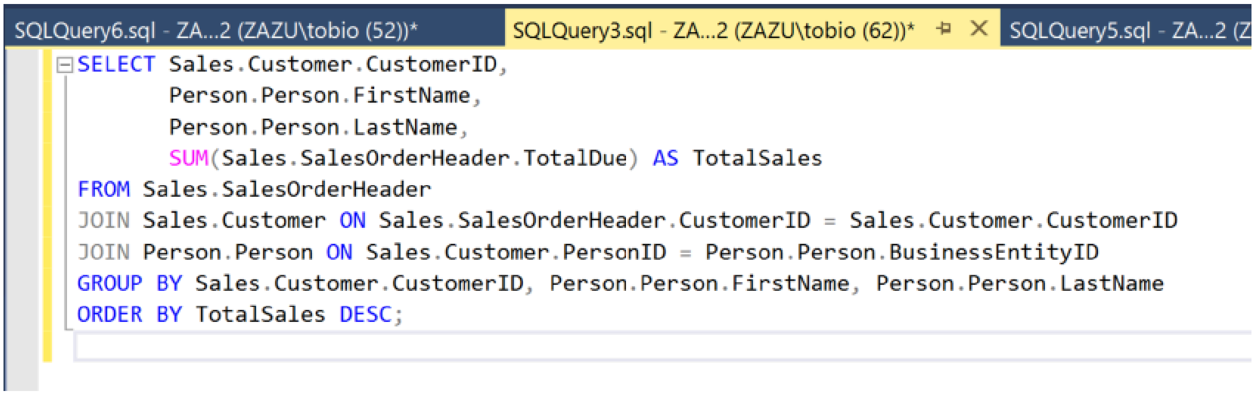
	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrd
1	43659	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
2	43660	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
3	43661	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
4	43662	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
5	43663	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
6	43664	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
7	43665	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0
8	43666	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate
1	1	NULL	934	1	AW00000001	3F5AE95E-B87D-4AED-95B4-C3797AFCB74F	2014-09-12 11:15:
2	2	NULL	1028	1	AW00000002	E552F657-A9AF-4A7D-A645-C429D6E02491	2014-09-12 11:15:
3	3	NULL	642	4	AW00000003	130774B1-DB21-4EF3-98C8-C104BCD6ED6D	2014-09-12 11:15:
4	4	NULL	932	4	AW00000004	FF862851-1DAA-4044-BE7C-3E85583C054D	2014-09-12 11:15:
5	5	NULL	1026	4	AW00000005	83905BDC-6F5E-4F71-B162-C98DA069F38A	2014-09-12 11:15:
6	6	NULL	644	4	AW00000006	1A92DF88-BFA2-467D-BD54-FCB9E647FDD7	2014-09-12 11:15:
7	7	NULL	930	1	AW00000007	03E9273E-B193-448E-9823-FE0C44AEED78	2014-09-12 11:15:
8	8	NULL	1024	5	AW00000008	801368B1-4323-4BFA-8BEA-5B5B1E4BD4A0	2014-09-12 11:15:

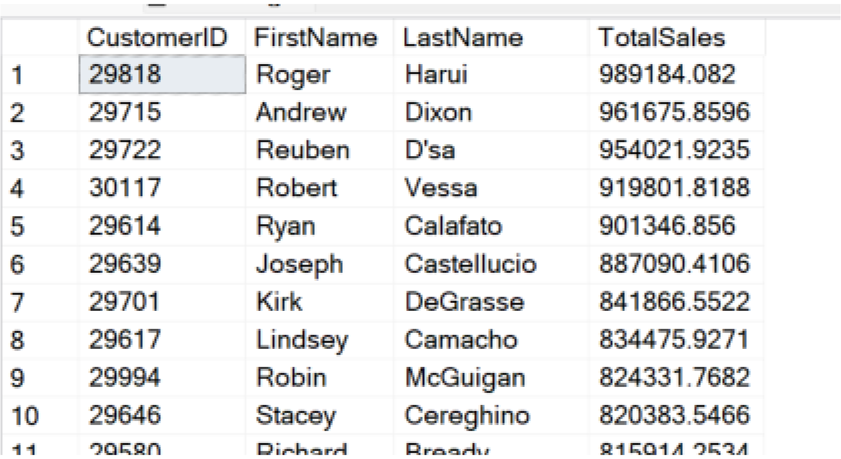
	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalC
1	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	NULL
2	2	EM	0	NULL	Terri	Lee	Duffy	NULL	1	NULL
3	3	EM	0	NULL	Roberto	NULL	Tamburello	NULL	0	NULL
4	4	EM	0	NULL	Rob	NULL	Walters	NULL	0	NULL
5	5	EM	0	Ms.	Gail	A	Erickson	NULL	0	NULL
6	6	EM	0	Mr.	Jossef	H	Goldberg	NULL	0	NULL
7	7	EM	0	NULL	Dylan	A	Miller	NULL	2	NULL
8	8	EM	0	NULL	Diane	L	Margheim	NULL	0	NULL

Query executed successfully.

From here i can run a query to join all three tables and calculate total sales for each customer


SQLQuery6.sql - ZA...2 (ZAZU\tobio (52))*
SQLQuery3.sql - ZA...2 (ZAZU\tobio (62))*
SQLQuery5.sql - ZA...2 (Z

```
SELECT Sales.Customer.CustomerID,
       Person.Person.FirstName,
       Person.Person.LastName,
       SUM(Sales.SalesOrderHeader.TotalDue) AS TotalSales
FROM Sales.SalesOrderHeader
JOIN Sales.Customer ON Sales.SalesOrderHeader.CustomerID = Sales.Customer.CustomerID
JOIN Person.Person ON Sales.Customer.PersonID = Person.Person.BusinessEntityID
GROUP BY Sales.Customer.CustomerID, Person.Person.FirstName, Person.Person.LastName
ORDER BY TotalSales DESC;
```



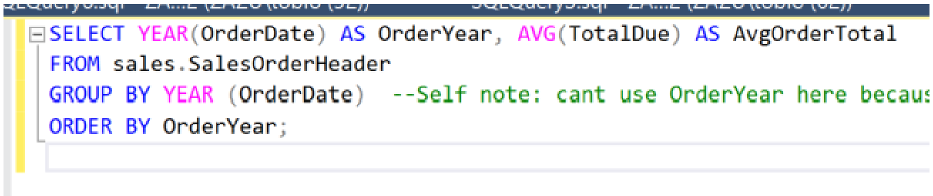
	CustomerID	FirstName	LastName	TotalSales
1	29818	Roger	Harui	989184.082
2	29715	Andrew	Dixon	961675.8596
3	29722	Reuben	D'sa	954021.9235
4	30117	Robert	Vessa	919801.8188
5	29614	Ryan	Calafato	901346.856
6	29639	Joseph	Castellucio	887090.4106
7	29701	Kirk	DeGrasse	841866.5522
8	29617	Lindsey	Camacho	834475.9271
9	29994	Robin	McGuigan	824331.7682
10	29646	Stacey	Cereghino	820383.5466
11	29580	Richard	Bready	815914.2534

A more experienced user would have used index to shorten the query but i did not index, because it's easier for me to read without them

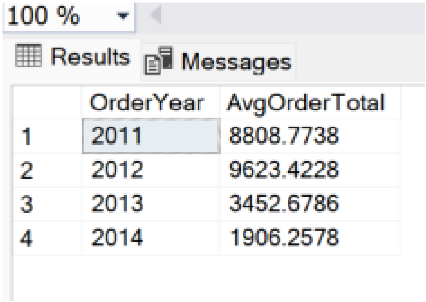
OBJECTIVES: Showcasing `JOINS` and `AGGREGATION`

Revenue Trends by Year

Now i want check out the average order revenue YOY to analyze trends



```
SELECT YEAR(OrderDate) AS OrderYear, AVG(TotalDue) AS AvgOrderTotal
FROM sales.SalesOrderHeader
GROUP BY YEAR (OrderDate)  --Self note: cant use OrderYear here becaus
ORDER BY OrderYear;
```



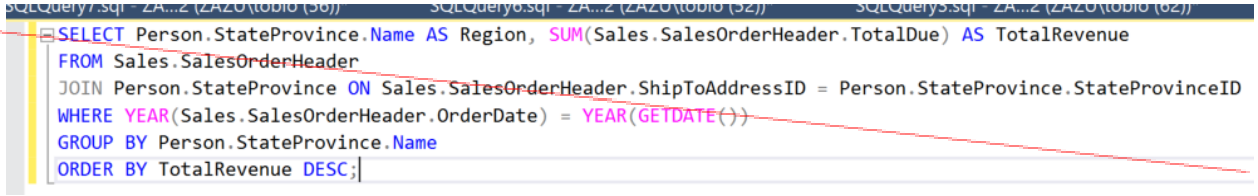
	OrderYear	AvgOrderTotal
1	2011	8808.7738
2	2012	9623.4228
3	2013	3452.6786
4	2014	1906.2578

By grouping orders by year, this query reveals trends in average order value over time, showcasing date functions and aggregation.

Showcasing `DATE functions` and `AGGREGATION`

Sales Revenue by Region

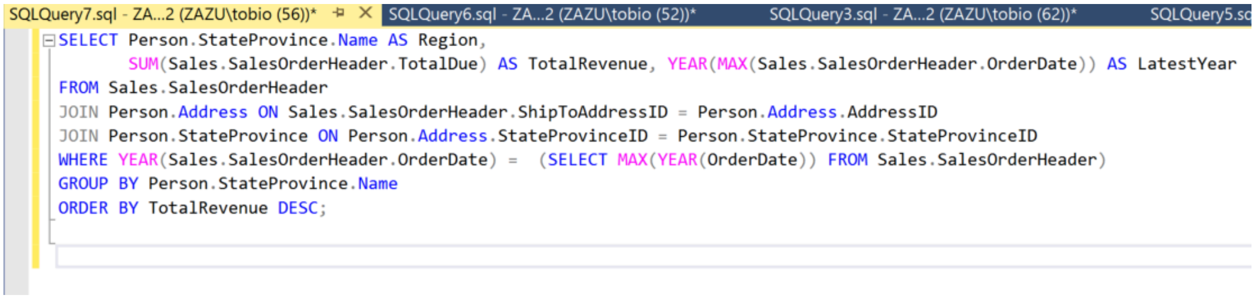
For the current year I want to calculate the total revenue by region. Using JOIN again I will pull data from *Sales.SalesOrderDetail* and *Production.Product* Table.



```
SELECT Person.StateProvince.Name AS Region, SUM(Sales.SalesOrderHeader.TotalDue) AS TotalRevenue
FROM Sales.SalesOrderHeader
JOIN Person.StateProvince ON Sales.SalesOrderHeader.ShipToAddressID = Person.StateProvince.StateProvinceID
WHERE YEAR(Sales.SalesOrderHeader.OrderDate) = YEAR(GETDATE())
GROUP BY Person.StateProvince.Name
ORDER BY TotalRevenue DESC;
```

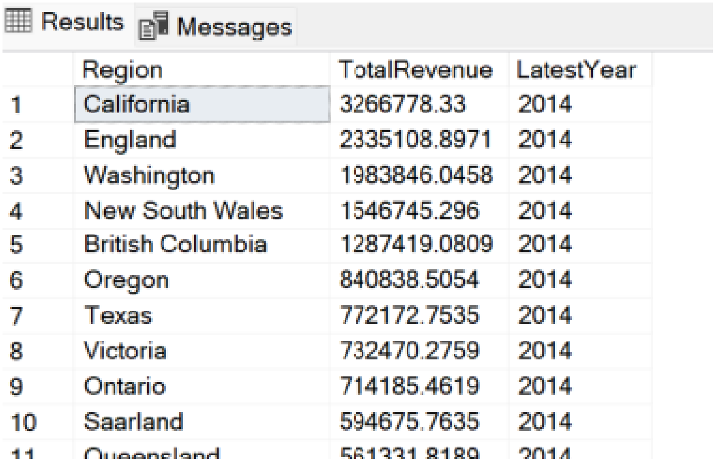
NOTES

At first, I assumed I could join SalesOrderHeader directly with StateProvince using ShipToAddressID, thinking it would provide a direct link to regional data. When the query did not return results, I took a step back to examine the database structure more closely. Through this process, I discovered that ShipToAddressID actually connects to AddressID in the Person.Address table, which then links to StateProvince. Additionally i was using the YEAR(GETDATE()) clause which gets the current year and since this data does not go up to 2024 it just returns as blank.


SQLQuery7.sql - ZA...2 (ZAZU\tobio (56))*
SQLQuery6.sql - ZA...2 (ZAZU\tobio (52))*
SQLQuery3.sql - ZA...2 (ZAZU\tobio (62))*
SQLQuery5.sq

```
SELECT Person.StateProvince.Name AS Region,
       SUM(Sales.SalesOrderHeader.TotalDue) AS TotalRevenue, YEAR(MAX(Sales.SalesOrderHeader.OrderDate)) AS LatestYear
FROM Sales.SalesOrderHeader
JOIN Person.Address ON Sales.SalesOrderHeader.ShipToAddressID = Person.Address.AddressID
JOIN Person.StateProvince ON Person.Address.StateProvinceID = Person.StateProvince.StateProvinceID
WHERE YEAR(Sales.SalesOrderHeader.OrderDate) =  (SELECT MAX(YEAR(OrderDate)) FROM Sales.SalesOrderHeader)
GROUP BY Person.StateProvince.Name
ORDER BY TotalRevenue DESC;
```

Here is the Query that I ended up with. I replaced GETDATE () with a MAX() function to fetch the latest date in the data.



	Region	TotalRevenue	LatestYear
1	California	3266778.33	2014
2	England	2335108.8971	2014
3	Washington	1983846.0458	2014
4	New South Wales	1546745.296	2014
5	British Columbia	1287419.0809	2014
6	Oregon	840838.5054	2014
7	Texas	772172.7535	2014
8	Victoria	732470.2759	2014
9	Ontario	714185.4619	2014
10	Saarland	594675.7635	2014
11	Queensland	561331.8189	2014

Showcasing

Joins: Combined data from multiple tables based on common fields.

Aggregations: Used `SUM`, `COUNT`, and `AVG` for data summarization.

Date Functions: Applied `YEAR()` and `GETDATE()` for time-based analysis.

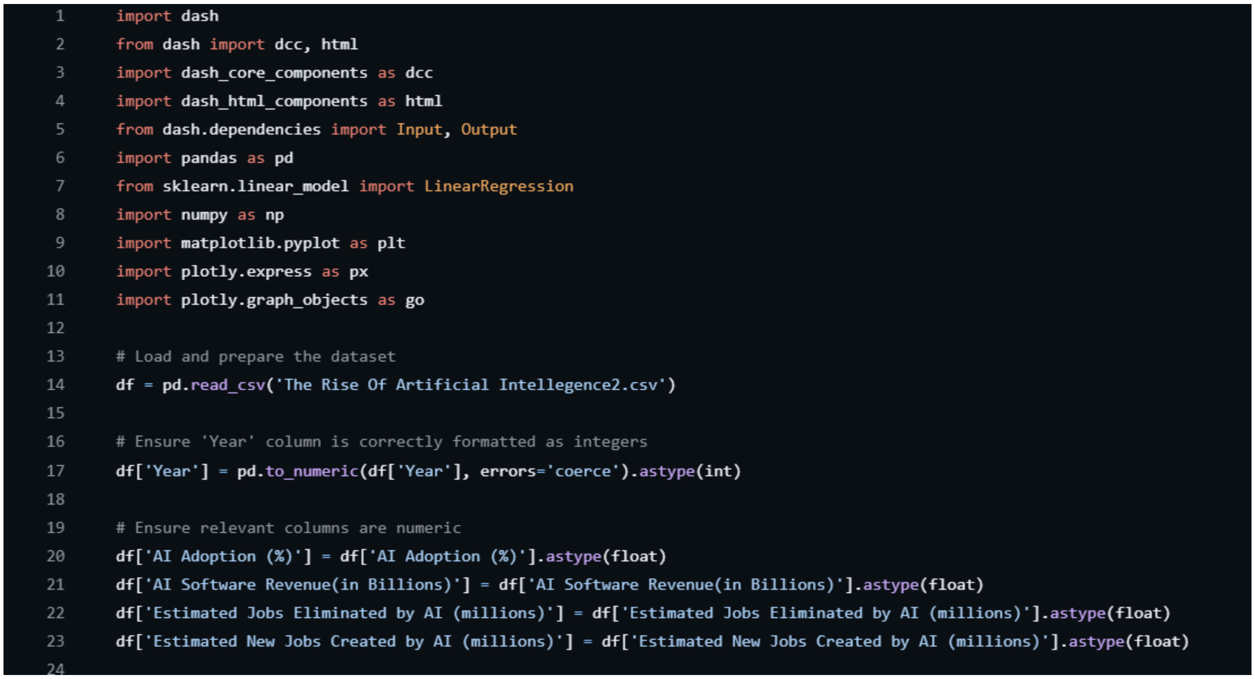
Python Portfolio

AI Trends Analysis Demonstrating Python Proficiency (Personal Project)

This project showcases my proficiency in navigating Python for data analysis and modeling. I utilized Python libraries to load, clean, and analyze a dataset on AI trends, applying correlation analysis and linear regression to generate insights.

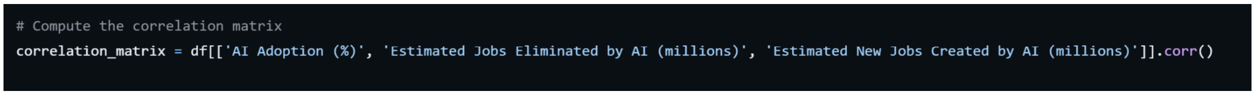
Data Loading and Cleaning

- **Library**: `pandas` for data manipulation.
- **Key Functions**: `pd.read_csv()` to load data, `pd.to_numeric()` for type conversion, `.astype()` to ensure compatibility for numerical analysis.
- Loaded CSV data, ensured column formats, and handled any data type issues to prepare for analysis.

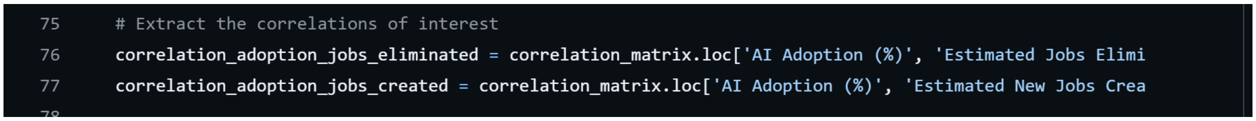


```python
import dash
from dash import dcc, html
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import pandas as pd
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

# Load and prepare the dataset
df = pd.read_csv('The Rise Of Artificial Intellegence2.csv')

# Ensure 'Year' column is correctly formatted as integers
df['Year'] = pd.to_numeric(df['Year'], errors='coerce').astype(int)

# Ensure relevant columns are numeric
df['AI Adoption (%)'] = df['AI Adoption (%)'].astype(float)
df['AI Software Revenue(in Billions)'] = df['AI Software Revenue(in Billions)'].astype(float)
df['Estimated Jobs Eliminated by AI (millions)'] = df['Estimated Jobs Eliminated by AI (millions)'].astype(float)
df['Estimated New Jobs Created by AI (millions)'] = df['Estimated New Jobs Created by AI (millions)'].astype(float)
```

Identified correlations and relationships between variables.

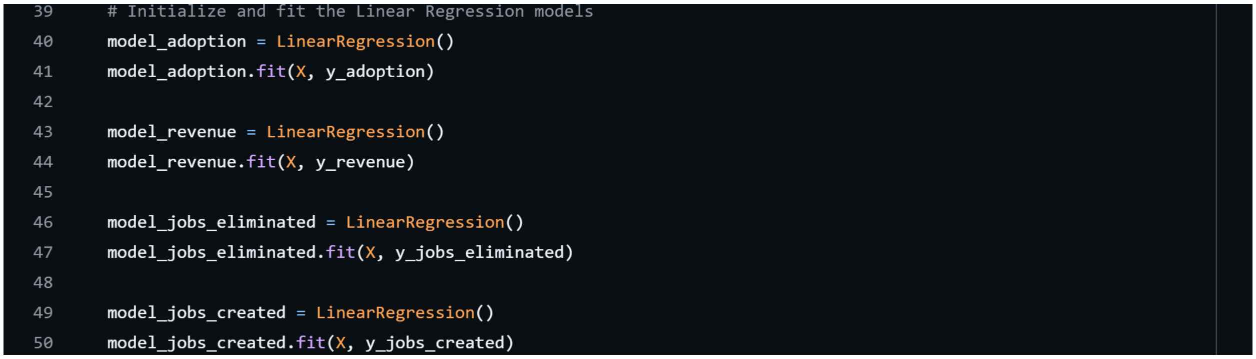


```python
# Compute the correlation matrix
correlation_matrix = df[['AI Adoption (%)', 'Estimated Jobs Eliminated by AI (millions)', 'Estimated New Jobs Created by AI (millions)']].corr()
```



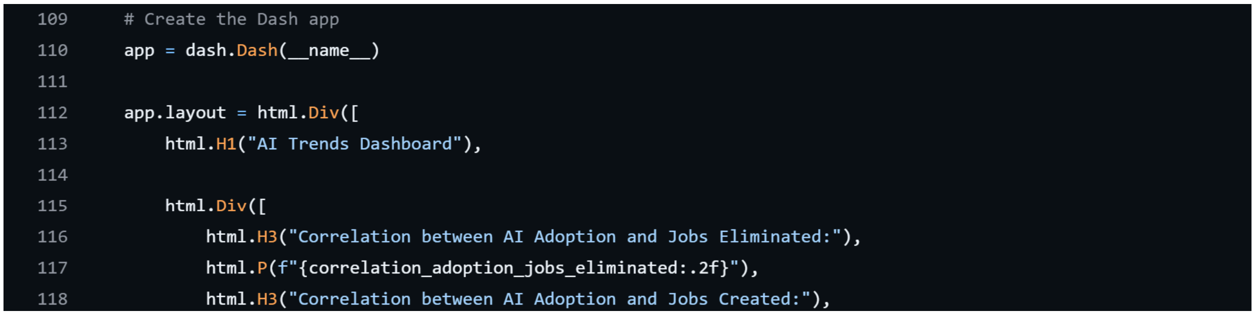
```python
# Extract the correlations of interest
correlation_adoption_jobs_eliminated = correlation_matrix.loc['AI Adoption (%)', 'Estimated Jobs Elimi
correlation_adoption_jobs_created = correlation_matrix.loc['AI Adoption (%)', 'Estimated New Jobs Crea
```

Predictive Modeling using SKLEARN Library for linear regression



```python
# Initialize and fit the Linear Regression models
model_adoption = LinearRegression()
model_adoption.fit(X, y_adoption)

model_revenue = LinearRegression()
model_revenue.fit(X, y_revenue)

model_jobs_eliminated = LinearRegression()
model_jobs_eliminated.fit(X, y_jobs_eliminated)

model_jobs_created = LinearRegression()
model_jobs_created.fit(X, y_jobs_created)
```

Experimented creating dashboards using Dash Library & Plotly for the first time instead of Matplotlib



```python
# Create the Dash app
app = dash.Dash(__name__)

app.layout = html.Div([
    html.H1("AI Trends Dashboard"),

    html.Div([
        html.H3("Correlation between AI Adoption and Jobs Eliminated:"),
        html.P(f"{correlation_adoption_jobs_eliminated:.2f}"),
        html.H3("Correlation between AI Adoption and Jobs Created:"),
```

Highlighting Python Proficiency

- **Libraries Learned**: `pandas`, `scikit-learn`, `plotly`, `Dash`.
- **Functions Used**: Applied functions like `.read_csv()`, `.astype()`, `.corr()`, `.fit()`, `.predict()`, `px.line()`, showcasing familiarity with essential tools in Python's data science ecosystem.
- **Workflow Efficiency**: Structured modular code to handle complex data, analysis, and visualizations.

There is still a lot i don't know in Python but i am a very fast learner

See full analysis on my github repository

Incident Handler Portfolio (Google Professional Cyber Security CERT.)

Portfolio Item I: Incident Handler's Journal – Ransomware Attack at Healthcare Clinic

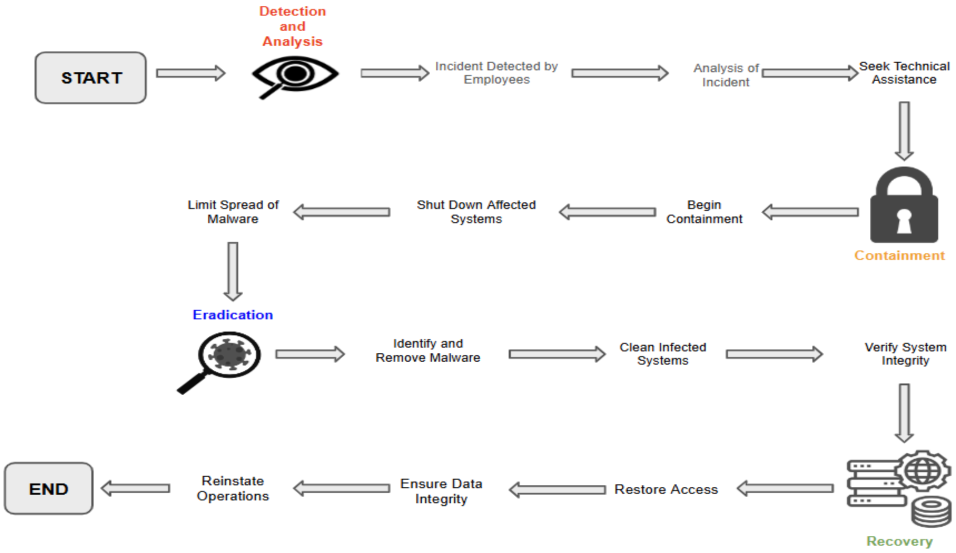
Overview: This entry documents a ransomware incident response scenario encountered by a U.S. healthcare clinic. The report outlines the incident detection, analysis, and containment measures taken, with insights into how the attack impacted business operations.

Incident Summary

- Who: Organized group of unethical hackers
- What: Ransomware incident disabling access to critical files and medical records
- Where: Healthcare clinic's IT systems
- When: Tuesday, 9:00 a.m.
- Why: Attackers exploited phishing vulnerabilities to gain access and deploy ransomware, seeking financial gain by demanding a ransom.

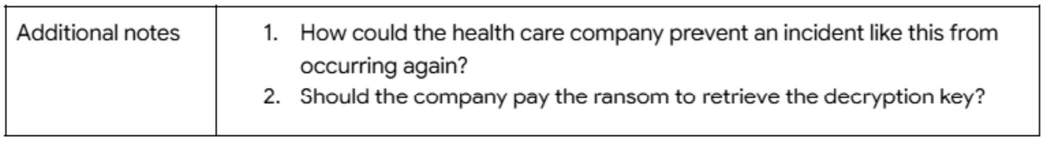
Response Phases

1. Detection and Analysis: Incident was detected when employees noticed file access issues and ransom notes. Technical assistance was sought from specialized organizations.
2. Containment, Eradication, and Recovery: Systems were shut down to prevent further spread. Since in-house resources were insufficient for recovery, external agencies were engaged for technical support.

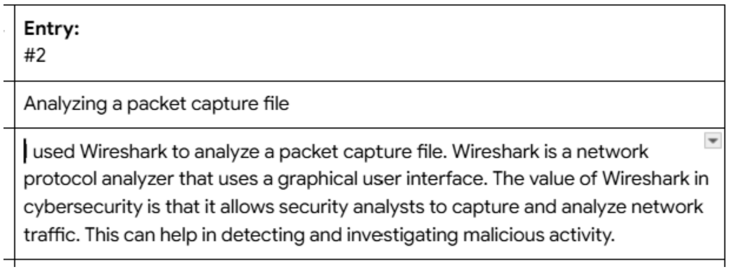


This flowchart shows the incident response process, from initial detection and containment through to eradication and recovery. Each step outlines critical actions taken to mitigate and resolve the ransomware incident.

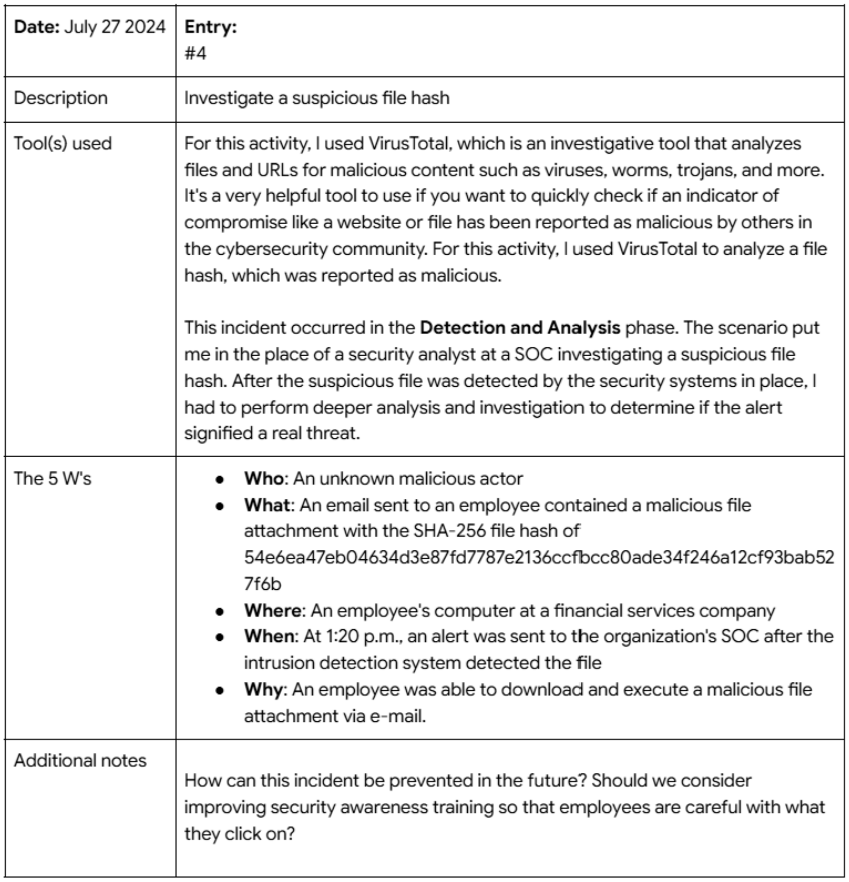
Detailed entries accompanying each step in the process



Additional notes	1. How could the health care company prevent an incident like this from occurring again? 2. Should the company pay the ransom to retrieve the decryption key?



Entry: #2
Analyzing a packet capture file
I used Wireshark to analyze a packet capture file. Wireshark is a network protocol analyzer that uses a graphical user interface. The value of Wireshark in cybersecurity is that it allows security analysts to capture and analyze network traffic. This can help in detecting and investigating malicious activity.



Date: July 27 2024	**Entry:** #4
Description	Investigate a suspicious file hash
Tool(s) used	For this activity, I used VirusTotal, which is an investigative tool that analyzes files and URLs for malicious content such as viruses, worms, trojans, and more. It's a very helpful tool to use if you want to quickly check if an indicator of compromise like a website or file has been reported as malicious by others in the cybersecurity community. For this activity, I used VirusTotal to analyze a file hash, which was reported as malicious. This incident occurred in the **Detection and Analysis** phase. The scenario put me in the place of a security analyst at a SOC investigating a suspicious file hash. After the suspicious file was detected by the security systems in place, I had to perform deeper analysis and investigation to determine if the alert signified a real threat.
The 5 W's	• **Who**: An unknown malicious actor • **What**: An email sent to an employee contained a malicious file attachment with the SHA-256 file hash of 54e6ea47eb04634d3e87fd7787e2136ccfbcc80ade34f246a12cf93bab527f6b • **Where**: An employee's computer at a financial services company • **When**: At 1:20 p.m., an alert was sent to the organization's SOC after the intrusion detection system detected the file • **Why**: An employee was able to download and execute a malicious file attachment via e-mail.
Additional notes	How can this incident be prevented in the future? Should we consider improving security awareness training so that employees are careful with what they click on?

Tools and Skills Used

- **Documentation**: Accurate and detailed incident logging was crucial.
- **Communication**: Collaboration with external organizations to secure technical support.

Lessons Learned

- **Preventative Measures**: Importance of phishing training and awareness to prevent similar incidents.
- **Debate on Ransom Payment**: Ethical and strategic considerations regarding ransom payment to regain data access.

Additional Reflections

- **Challenges**: This incident underscored the complexity of cybersecurity threats and the need for robust response plans.

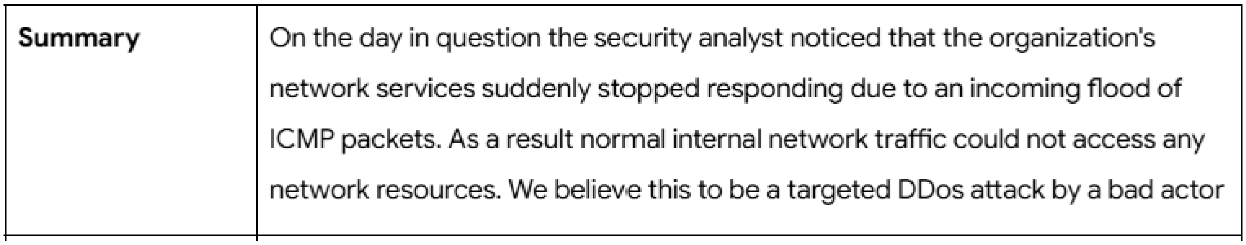
- **Growth**: Enhanced understanding of the end-to-end response process, from initial detection to containment and recovery.

Portfolio Item II: Incident Handler's Journal – DDoS Attack at Corporate Network

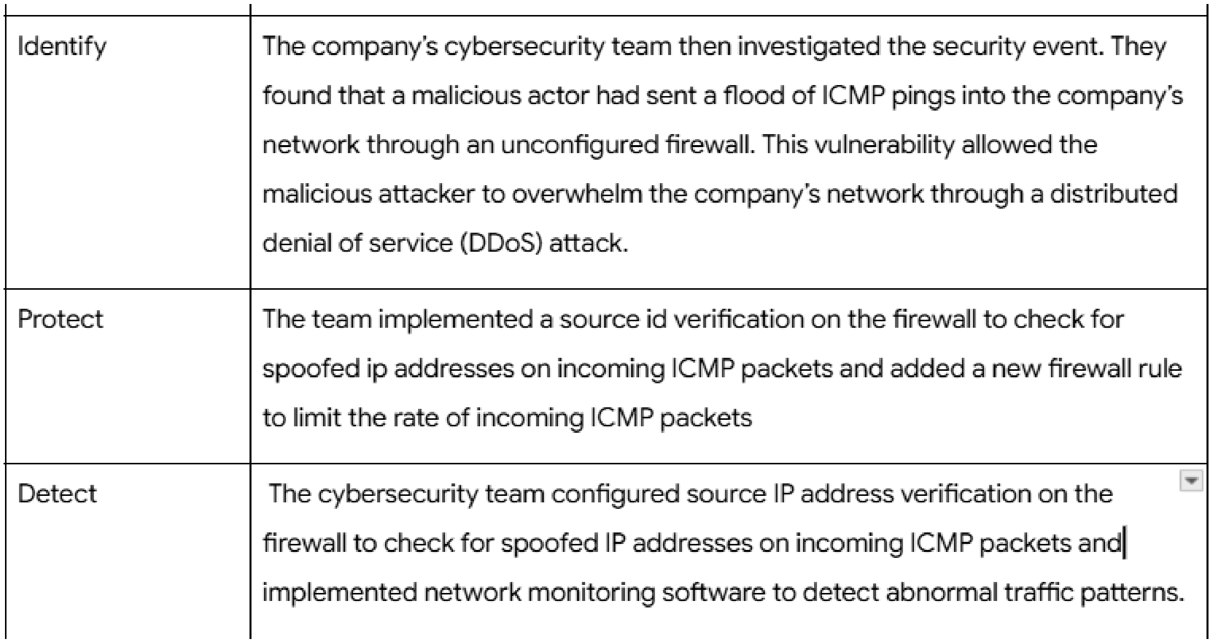
This entry documents a Distributed Denial of Service (DDoS) incident where an organization's network services were disrupted due to a flood of ICMP packets. This report follows the NIST framework, covering each phase from identification to recovery.

Incident Summary

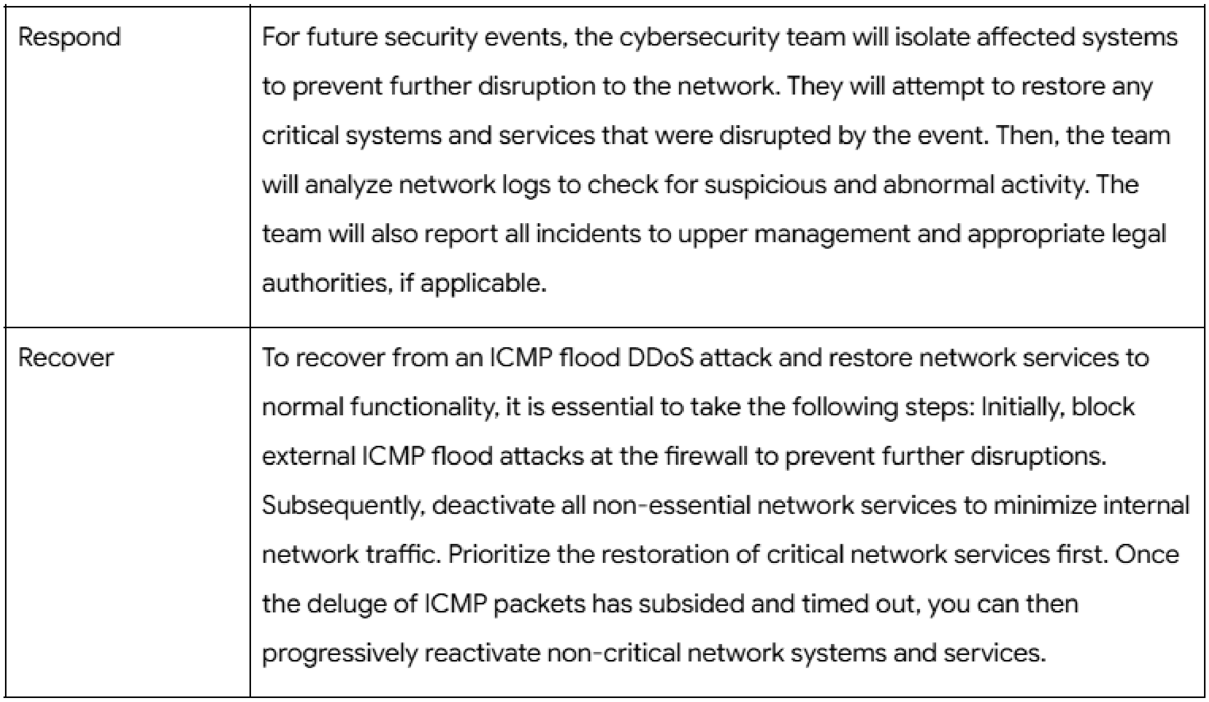
- **What**: DDoS attack disabling network access
- **Where**: Corporate network services
- **When**: During regular business operations
- **How**: Flood of ICMP packets overwhelmed the network, exploiting an unconfigured firewall



Summary	On the day in question the security analyst noticed that the organization's network services suddenly stopped responding due to an incoming flood of ICMP packets. As a result normal internal network traffic could not access any network resources. We believe this to be a targeted DDos attack by a bad actor



Identify	The company's cybersecurity team then investigated the security event. They found that a malicious actor had sent a flood of ICMP pings into the company's network through an unconfigured firewall. This vulnerability allowed the malicious attacker to overwhelm the company's network through a distributed denial of service (DDoS) attack.
Protect	The team implemented a source id verification on the firewall to check for spoofed ip addresses on incoming ICMP packets and added a new firewall rule to limit the rate of incoming ICMP packets
Detect	The cybersecurity team configured source IP address verification on the firewall to check for spoofed IP addresses on incoming ICMP packets and implemented network monitoring software to detect abnormal traffic patterns.



Respond	For future security events, the cybersecurity team will isolate affected systems to prevent further disruption to the network. They will attempt to restore any critical systems and services that were disrupted by the event. Then, the team will analyze network logs to check for suspicious and abnormal activity. The team will also report all incidents to upper management and appropriate legal authorities, if applicable.
Recover	To recover from an ICMP flood DDoS attack and restore network services to normal functionality, it is essential to take the following steps: Initially, block external ICMP flood attacks at the firewall to prevent further disruptions. Subsequently, deactivate all non-essential network services to minimize internal network traffic. Prioritize the restoration of critical network services first. Once the deluge of ICMP packets has subsided and timed out, you can then progressively reactivate non-critical network systems and services.

Lessons Learned

- **Firewall Hardening**: This incident highlighted the importance of proactive firewall configuration and IP address filtering.
- **Incident Preparedness**: The response plan was updated to include more rapid detection and response measures for similar attacks.

Showcased Skills

- Protocols: TCP/IP, (Familiarity)
- Tools: Wireshark, tcpdump, VirusTotal
- Incident Response Frameworks: NIST, custom organizational protocols
- Network Security: IP filtering, firewall configuration, network traffic monitoring

Thank you for taking the time to explore my portfolio.

Each project in this portfolio reflects my dedication to continuous learning and my passion for solving complex technical challenges.

I look forward to discussing how my background and skill set align with your company's goals and how I can continuously bring value to your team.

Github:https://github.com/Quoe102