GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT127-3-2-PFDA-T-36

PROGRAMMING FOR DATA ANALYSIS

GROUP NUMBER: 29

HAND OUTDATE: 6 JANUARY 2025

HAND IN DATE: 17 FEBRUARY 2025

LECTURER NAME: MRS. FARHANA ILLIANI BINTI HASSAN

No.	Student Name	TP Number	Signature
1.	Manreen Kaur A/P Jagjit Singh	TP071290	*Mrn*
2.	Lim Eazen	TP078935	*Eazen*
3.	Brayden Yoong Policarpio	TP079094	*Bray*
4.	Ong Kai Ying	TP086065	*KaiYing*

1.0 Introduction 6
1.1 Data Description 6
1.2 Assumptions 6
1.3 Techniques 6
1.3 Hypothesis and Objectives 7
2.0 Data Preparation 8
2.1 Data Import 8
2.2 Data Preprocessing and Cleaning 8
2.3 Data Validation 9
3.0 Data Analysis 11
3.1 Objective 1: To analyze trends in web hacking activities by examining web server distributions, ransom attack occurrences, and downtime across different countries. The goal is to identify which web servers and countries experience the most attacks and prolonged downtime. - Manreen Kaur A/P Jagjit Singh (TP071290) 11
3.1.1 Analysis 1-1.2: Lollipop Chart: The distribution of Top 5 Webserver based on Count and Percentage (3D Graph) 12
3.1.2 Analysis 1-2: Bar Chart: The distribution of Top 10 Webserver across Ransom Attacks 13
3.1.3 Analysis 1-3: Histogram: The distribution of Top 10 Countries across Ransom Attacks 14
3.1.5 Analysis 1-5: Bar Chart with Line Graph: Web Servers and Ransom Attacks based on Downtime .. 16
3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph: Web Server, Ransom Attacks, Downtime, and Country. 18
Conclusion based on 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph: Web Server, Ransom Attacks, Downtime, and Country. 20
Additional Feature 22
3.2 Objective 2: The study analyzes web server resilience to ransom attacks by examining downtime, attack frequency, and notification impact across countries (Lim Eazen TP078935) 24
3.2.1 Which web servers are most frequently attacked, and what does this reveal about their resilience against cyber threats? 24
3.2.4 How does the frequency of attacks on different web servers correlate with their downtime, and which servers demonstrate the strongest resilience against frequent cyberattacks? 29
3.2.5 How does the frequency of ransom attacks across different countries impact the downtime of web servers, and which countries have the fastest recovery times? 31

This analysis confirms that attack frequency does not directly correlate with downtime. Some servers recover quickly, while others suffer from extended outages. Identifying these patterns helps improve security measures and recovery pprocesses. 33

Conclusion 33

Key Findings 33

Final Insights 34

3.3 Objective 3: To examine the correlation between web server types, operating systems, and ransom attack severity by analyzing attack patterns, downtime, and financial losses. 35

3.3.1 Analysis 1: Examining the relationship between OS and Web Server Vulnerability 35

3.3.2 Analysis 2: Examining the relationship between OS, Web Server & Downtime 36

3.3.3 Analysis 3: Examining the relationship between OS, Web Server & Financial Losses 37

3.3.4 Analysis 4: Examining the relationship between OS, Web Server & Attack Trends Over Time 38

3.3.5 Analysis 4: Examining the relationship between OS, Web Server, Ransom Attacks & Country-Specific Trends 39

3.3.6 Conclusion 40

3.4 Objective 4: To analyze the trends of hacking activities in recent years and identify the most prevalent attack types, along with their associated vulnerabilities and impacts. (Ong Kai Ying TP086065) 41

3.4.1 What causes fluctuations in hacking incidents since 2003, and how do unique IP trends reflect changes in attacker behavior? 41

3.4.2 Which operating systems face the highest attack frequency, and which OS has the most ransomware incidents and the highest financial loss? 35

3.4.3 Are there specific IPs that repeatedly attack within a short period, indicating potential DDoS or brute-force attempts? 44

3.4.4 Which web servers are most frequently targeted in attacks, and what factors contribute to the varying ransom demands and financial losses associated with these attacks? 47

Additional Feature 50

3.4.5 Which country experiences the largest financial losses due to ransom demands? 50

Conclusion 53

Workload Matrix 54

Reference 55

Figure 1: 2.1 Data Import (Install & Load libraries) 8
Figure 2: 2.1 Data Import (Import Original Data)
Figure 3: 2.1 Import Data (Inspect data structure) 8
Figure 4: 2.2 Data Processing and Cleaning (Convert column numeric, remove special characters) 8
Figure 5:2.2 Data Processing and Cleaning (NA & Empty values replace with mean) 8
Figure 6:2.2 Data Processing and Cleaning (Missing value replace with mean) 8
Figure 7:2.2 Data Processing and Cleaning (Empty categorical value replace with most frequent value) 8
Figure 8:2.2 Data Processing and Cleaning (Convert date format)
Figure 9:2.2 Data Processing and Cleaning (Valid country name list) 8
Figure 10:2.2 Data Processing and Cleaning (Data of wrong country name and missing space) 9
Figure 11:2.2 Data Processing and Cleaning (Find closest country name and correct country name) 9
Figure 12:2.2 Data Processing and Cleaning (Dictionary country mapping to language & Uppercase country name to match dictionary) 9
Figure 13:2.2 Data Processing and Cleaning (Fix spacing)
Figure 14:2.2 Data Processing and Cleaning (Map country to correct language) 9
Figure 15:2.2 Data Processing and Cleaning (Replace missing language values with most frequent value) 9
Figure 16:2.2 Data Processing and Cleaning(All text Uppercase & remove duplicate)
2.3 Data Validation 9
Figure 17: 2.3 Data Validation (Make outlier & replace outlier with mean) 9
Figure 18: 2.3 Data Validation (Check column name exist)
Figure 19: 2.3 Data Validation (Check missing values & rows) 10
Figure 20: 2.3 Data Validation (Outlier detection if column exist)
Figure 21: 2.3 Data Validation (Replace missing value with mean) 10
Figure 22: 2.3 Data Validation (No negative / 0 value)
Figure 23: 2.3 Data Validation (Column numbers exist before make function) 10
Figure 24: 2.3 Data Validation (Downtime summary)
Figure 25: 2.3 Data Validation (Rename downtime column) 10
Figure 26: 2.3 Data Validation (Convert downtime & ransom numeric)
Figure 27: 2.3 Data Validation (Display downtime & ransom) 10
Figure 28: 2.3 Data Validation (Correct country display in data)
Figure 29: 2.3 Data Validation (Count country in descending) 10
Figure 30: 2.3 Data Validation (Save clean data)
Figure 31: 2.3 Data Validation (Display data)
Figure 32: 2.3 Data Validation (Run csv file) 10
Figure 33: 3.1.1 Analysis 1-1.2 Lollipop Chart Code (2D) 11
Figure 34: 3.1.1 Analysis 1-1.2 Lollipop Chart (2D) 12
Figure 35: 3.1.1 Analysis 1-1.2: Lollipop Chart Code (3D)
Figure 36: 3.1.1 Analysis 1-1.2: Lollipop Chart (3D) 12
Figure 37: 3.1.2 Analysis 1-2: Bar Chart Code 13
Figure 38: 3.1.2 Analysis 1-2: Bar Chart 13

Figure 39: 3.1.3 Analysis 1-3: Histogram Code
Figure 40: 3.1.3 Analysis 1-3: Histogram (Table Output)....14
Figure 41: 3.1.3 Analysis 1-3: Histogram Chart....14
Figure 42: 3.1.4 Analysis 1-4: Pie Chart Code
Figure 43: 3.1.4 Analysis 1-4: Pie Chart (Table output)....15
Figure 44: 3.1.4 Analysis 1-4: Pie Chart....15
Figure 45: 3.1.5 Analysis 1-5: Bar Chart with Line Graph Code....16
Figure 46: 3.1.5 Analysis 1-5 (Table Output)....16
Figure 47: 3.1.5 Analysis 1-5: Bar Chart with Line Graph....17
Figure 48: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph Code....18
Figure 49: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 1)....18
Figure 50: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 2)....18
Figure 51: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 3)....18
Figure 52: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Print Total Rows)....18
Figure 53: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph....19
Figure 54: Conclusion (Chi-Square Test)....20
Figure 55: Conclusion (T-Test)....20
Figure 56: Conclusion (Correlation Test)....21

1.0 Introduction

1.1 Data Description

In this assignment, the team conducted an in-depth data analysis on a dataset containing information about web hacking cases over the past 15 years. This dataset includes various attributes such as attack date, notifier, IP address, web server, and other relevant details related to cyberattacks. Before conducting the analysis, the dataset was thoroughly cleaned, refactored, and sub-setted to ensure data integrity and accuracy.

The objective of this study is to identify hacking trends and provide meaningful recommendations to stakeholders. Different variables from the dataset were selected to test hypotheses related to hacking incidents. The selected variables were analyzed and visualized using R programming techniques to determine their impact on cyberattack trends and patterns.

1.2 Assumptions

In this assignment, each group member selected one key variable to observe and analyze in order to assess its impact on hacking trends. After conducting individual analyses, the group combined their selected variables for a joint analysis.

The final analysis consists of four combined variables that were examined collectively to derive insights into hacking activities. The group ensured that the conclusions drawn from individual analyses aligned with the overall findings. If the combined variables did not present a strong enough correlation, further exploration of additional variables was conducted to demonstrate the significance of hacking trends and patterns.

1.3 Techniques

The dataset was analyzed using various data analysis techniques, including:

- **Data Exploration** – Understanding the structure and nature of the dataset.
- **Data Manipulation** – Cleaning, transforming, and refining the data for analysis.

- **Data Transformation** – Formatting and restructuring data for better interpretation.
- **Data Visualization** – Graphical representation of hacking trends using R programming.

To enhance the effectiveness of data retrieval, additional advanced techniques were explored beyond those covered in the course. These included trend analysis, anomaly detection, and correlation analysis.

1.3 Hypothesis and Objectives

Hypothesis:

There is a significant relationship between the country, web server, ransom attack frequency, and downtime, with some countries and web servers experiencing more frequent attacks and prolonged downtime. Furthermore, the frequency and duration of these attacks are expected to vary based on the web server type and the implemented security protocols (such as firewalls, intrusion detection systems, or patch management practices), which may impact the resilience of the servers in recovering from these attacks.

Objectives:

1. Impact of web server distributions, ransom attack occurrences, and downtime across countries on web hacking trends.
2. To assess the resilience of web servers against ransom attacks by analyzing attack frequency, downtime, and notification strategies across different countries. The goal is to identify which web servers recover fastest, which countries face the most downtime, and how notification methods influence security responses.
3. To examine the correlation between web server types, operating systems, and ransom attack severity by analyzing attack patterns, downtime, and financial losses.
4. To analyze the trends of hacking activities in recent years and identify the most prevalent attack types, along with their associated vulnerabilities and impacts.

2.0 Data Preparation

2.1 Data Import

```
# 1. Install & Load Required libraries
# Install the libraries
install.packages("stringr")     # Install stringr for string processing
install.packages("ggplot2")     # Install ggplot2 for visualization
install.packages("ggpubr", dependencies = TRUE)
install.packages("corrplot", dependencies = TRUE)
install.packages("reshape2", dependencies = TRUE)
install.packages("ggridges")
install.packages("dplyr")      # Install dplyr if not already installed
install.packages("tidyr")      # Install tidyr for data manipulation
install.packages("stringdist")
install.packages("ggrepel")
install.packages("plotly")
if (!requireNamespace("countrycode", quietly = TRUE)) install.packages("countrycode")
install.packages("knitr")  # For kable to display table
install.packages("DT")     # For interactive tables
install.packages("tidyverse")
install.packages("readr")
```

```
# Load the Libraries
library(stringr)
library(ggplot2)
library(ggpubr)
library(corrplot)
library(reshape2)
library(ggridges)
library(dplyr)
library(tidyr)
library(stringdist)
library(ggrepel)
library(plotly)
library(countrycode)
library(knitr)
library(DT)
library(tidyverse)
library(readr)
```

Figure 1: 2.1 Data Import (Install & Load libraries)

```
# 2. Import original data
setwd("C:/Users/Manreen/OneDrive/Documents/R PFDA ASSIGNMENT")
data <- read.csv("4.hackingdata.csv")
```

Figure 2: 2.1 Data Import (Import Original Data)

```
# 3. Inspect data structure
str(data)   # Check the structure of the dataset
dim(data)   # Check the number of rows and columns
head(data)  # View the first few rows
summary(data)
```

Figure 3: 2.1 Import Data (Inspect data structure)

2.2 Data Preprocessing and Cleaning

```
# Convert relevant columns to numeric, removing special characters
colnames(data)
names(data) <- trimws(names(data))  # Remove extra spaces
names(data) <- tolower(names(data)) # Convert to lowercase for consistency
data <- read.csv("4.hackingdata.csv")

data <- data %>%
  mutate(
    Ransom = as.numeric(gsub("[^0-9.]", "", as.character(Ransom))),
    Loss = as.numeric(gsub("[^0-9.]", "", as.character(Loss)))
  )
```

Figure 4: 2.2 Data Processing and Cleaning (Convert column numeric, remove special characters)

```
# Replace only NA and empty values in Ransom with the mean, keeping existing values unchanged
if ("Ransom" %in% colnames(data)) {
  ransom_mean <- mean(data$Ransom, na.rm = TRUE)  # Calculate mean excluding NAs
  data$Ransom[is.na(data$Ransom) | data$Ransom == ""] <- ransom_mean  # Replace only NA and empty values
}
```

Figure 5:2.2 Data Processing and Cleaning (NA & Empty values replace with mean)

```
# Handle missing values by replacing them with the mean
num_cols <- sapply(data, is.numeric)
data[, num_cols] <- lapply(data[, num_cols], function(col) {
  ifelse(is.na(col), mean(col, na.rm = TRUE), col)
})
```

Figure 6:2.2 Data Processing and Cleaning (Missing value replace with mean)

```
# Fill empty categorical values with the most frequent value
fill_most_frequent <- function(column) {
  column[column == ""] <- NA
  column[is.na(column)] <- names(which.max(table(column)))
  return(column)
}

data$Lang <- fill_most_frequent(data$Lang)
data$WebServer <- fill_most_frequent(data$WebServer)
data$Country <- fill_most_frequent(data$Country)
```

Figure 7:2.2 Data Processing and Cleaning (Empty categorical value replace with most frequent value)

```
# Convert Date column to proper format
data$Date <- as.Date(data$Date, format = "%d/%m/%Y")
```

Figure 8:2.2 Data Processing and Cleaning (Convert date format)

```
# Get a list of valid country names
Country <- unique(countrycode::codelist$country.name.en)
```

Figure 9:2.2 Data Processing and Cleaning (Valid country name list)

```r
# Sample data with incorrect country names and missing spaces
df <- data.frame(
  Country = c("Untied States", "UNITED STATE", "Germnay", "Barzil", "RUSSIAN FEDE",
              "UNITED KINGD", "NETHERLANDS\"", "NEWZEALAND", "PUERTORICO", "CZECH REPUBL",
              "SOUTHAMERICA", "VIET NAM"),
  count = c(1555, 257, 6908, 6625, 68, 257, 75, 77, 10, 23, 100, 200)
)
```

Figure 10:2.2 Data Processing and Cleaning (Data of wrong country name and missing space)

```r
# Function to find the closest correct country name
find_closest_match <- function(country) {
  distances <- stringdist::stringdist(country, Country, method = "jw")  # Jaro-Winkler similarity
  return(Country[which.min(distances)])  # Return the closest match
}

# Apply function to correct country names
df$Country <- sapply(df$Country, find_closest_match)
```

Figure 11:2.2 Data Processing and Cleaning (Find closest country name and correct country name)

```r
# Define a dictionary mapping countries to their primary languages
Country_to_lang <- c(
  "UNITED STATES" = "English",
  "GERMANY" = "German",
  "BRAZIL" = "Portuguese",
  "RUSSIA" = "Russian",
  "UNITED KINGDOM" = "English",
  "NETHERLANDS" = "Dutch",
  "NEW ZEALAND" = "English",
  "PUERTO RICO" = "Spanish",
  "CZECH REPUBLIC" = "Czech",
  "SOUTH AMERICA" = "Spanish",
  "VIETNAM" = "Vietnamese"
)

# Ensure country names are in uppercase to match the dictionary keys
data$Country <- toupper(data$Country)
```

Figure 12:2.2 Data Processing and Cleaning (Dictionary country mapping to language & Uppercase country name to match dictionary)

```r
# Fix spacing issues (e.g., SOUTHAMERICA → SOUTH AMERICA)
df$Country <- gsub("([A-Z])([A-Z]+)", "\\1 \\2", df$Country)
```

Figure 13:2.2 Data Processing and Cleaning (Fix spacing)

```r
# Map corrected country names to languages
df$lang <- Country_to_lang[df$Country]
```

Figure 14:2.2 Data Processing and Cleaning (Map country to correct language)

```r
# Handle missing language values by assigning the most frequent language
if (sum(!is.na(df$lang)) > 0) {  # Check if there are non-NA values
  most_frequent_lang <- names(which.max(table(na.omit(df$lang))))  # Get most frequent language
  df$lang[is.na(df$lang)] <- most_frequent_lang  # Assign to missing values
} else {
  df$lang[is.na(df$lang)] <- "UNKNOWN"  # Fallback if all values are NA
}
```

Figure 15:2.2 Data Processing and Cleaning (Replace missing language values with most frequent value)

```r
# Convert all text to uppercase
df$Country <- toupper(df$Country)
df$lang <- toupper(df$lang)

# Remove duplicate entries
data <- unique(data)
```

Figure 16: 2.2 Data Processing and Cleaning(All text Uppercase & remove duplicate)

2.3 Data Validation

```r
# Define outlier detection function first
outlier_detection <- function(column) {
  if (sum(!is.na(column)) == 0) {  # If all values are NA, return column unchanged
    return(column)
  }
  Q1 <- quantile(column, 0.25, na.rm = TRUE)
  Q3 <- quantile(column, 0.75, na.rm = TRUE)
  mean <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * mean
  upper_bound <- Q3 + 1.5 * mean
  column[column < lower_bound | column > upper_bound] <- NA

# Replace outliers with mean to maintain the same length
  return(ifelse(is.na(column), mean(column, na.rm = TRUE), column))
}
```

Figure 17: 2.3 Data Validation (Make outlier & replace outlier with mean)

```r
# Ensure column names exist
colnames(data)
```

Figure 18: 2.3 Data Validation (Check column name exist)

```r
# Check missing values and total rows
sum(is.na(data$Downtime))  # Count NA values
length(data$Downtime)
sum(is.na(data$Ransom))    # Count NA values
length(data$Downtime)
```

Figure 19: 2.3 Data Validation (Check missing values & rows)

```r
# Apply outlier detection only if the column exists
if ("Downtime" %in% colnames(data)) {
  data$Downtime <- outlier_detection(data$Downtime)
}
```

Figure 20: 2.3 Data Validation (Outlier detection if column exist)

```r
# Replace missing values with mean
if ("Downtime" %in% colnames(data)) {
  data$Downtime[is.na(data$Downtime)] <- mean(data$Downtime, na.rm = TRUE)
}

if ("Ransom" %in% colnames(data)) {
  data$Ransom[is.na(data$Ransom)] <- mean(data$Ransom, na.rm = TRUE)
}
```

Figure 21: 2.3 Data Validation (Replace missing value with mean)

```r
# Ensure no negative or zero values
if ("Ransom" %in% colnames(data)) {
  data$Ransom[data$Ransom <= 0 | is.na(data$Ransom)] <- NA
}

if ("Downtime" %in% colnames(data)) {
  data$Downtime[data$Downtime <= 0 | is.na(data$Downtime)] <- NA
}
```

Figure 22: 2.3 Data Validation (No negative / 0 value)

```r
# Ensure num_cols exists before applying the function
if (exists("num_cols")) {
  data[, num_cols] <- lapply(data[, num_cols], function(col) {
    ifelse(is.na(col), mean(col, na.rm = TRUE), col)
  })
}
```

Figure 23: 2.3 Data Validation (Column numbers exist before make function)

```r
# Check summary of Downtime to confirm no errors
summary(data$Downtime)
```

Figure 24: 2.3 Data Validation (Downtime summary)

```r
# Rename column if necessary
colnames(data)[colnames(data) == "DownTime"] <- "Downtime"
```

Figure 25: 2.3 Data Validation (Rename downtime column)

```r
# Convert Downtime and Ransom to numeric, removing non-numeric characters
data$Downtime <- as.numeric(gsub("[^0-9.]", "", as.character(data$Downtime)))
data$Ransom <- as.numeric(gsub("[^0-9.]", "", as.character(data$Ransom)))
```

Figure 26: 2.3 Data Validation (Convert downtime & ransom numeric)

```r
# Print summary to verify transformations
summary(data$Downtime)
summary(data$Ransom)
```

Figure 27: 2.3 Data Validation (Display downtime & ransom)

```r
# Ensure corrected_country exists in data
if (!"Country" %in% colnames(data)) {
  data$Country <- toupper(data$Country)  # Use existing 'Country' column
}
```

Figure 28: 2.3 Data Validation (Correct country display in data)

```r
# Now count occurrences and sort
data %>%
  count(Country) %>%
  arrange(desc(n))
```

Figure 29: 2.3 Data Validation (Count country in descending)

```r
# Saved cleaned data in the same csv file
write.csv(data, "4.hackingdata.csv", row.names = FALSE)
```

Figure 30: 2.3 Data Validation (Save clean data)

```r
# Display summary statistics
summary(data)
```

Figure 31: 2.3 Data Validation (Display data)

```r
# After saving data.csv, verify it by running
new_data <- read.csv("4.hackingdata.csv")
str(new_data)
head(new_data)
```

Figure 32: 2.3 Data Validation (Run csv file)

3.0 Data Analysis

3.1 Objective 1: To analyze trends in web hacking activities by examining web server distributions, ransom attack occurrences, and downtime across different countries. The goal is to identify which web servers and countries experience the most attacks and prolonged downtime. - Manreen Kaur A/P Jagjit Singh (TP071290)

3.1.1 Analysis 1-1.1: Lollipop Chart: The distribution of Top 10 Webserver based on Count and Percentage (2D Graph)

```
# Create the lollipop chart
p <- ggplot(webserver_counts, aes(x = reorder(WebServer, Count), y = Count)) +
  geom_segment(aes(xend = WebServer, yend = 0), color = "black") +
  geom_point(color = "darkmagenta", size = 4) +
  geom_text(aes(label = paste0(Count, " (", round(Percentage, 1), "%)")),
            hjust = -0.2, size = 2.5, check_overlap = TRUE) +
  coord_flip() +
  scale_y_continuous(breaks = seq(0, 125000, by = 25000), expand = expansion(mult = c(0, 0.1))) +
  labs(title = "Web Server Distribution - Lollipop Chart", x = "Web Server", y = "Count") +
  theme_minimal() +
  theme(plot.margin = margin(10, 10, 10, 10),
        axis.line = element_line(color = "black"))  # Make x and y axis visible

# Print the plot
print(p)
```

Figure 33: 3.1.1 Analysis 1-1.2 Lollipop Chart Code (2D)

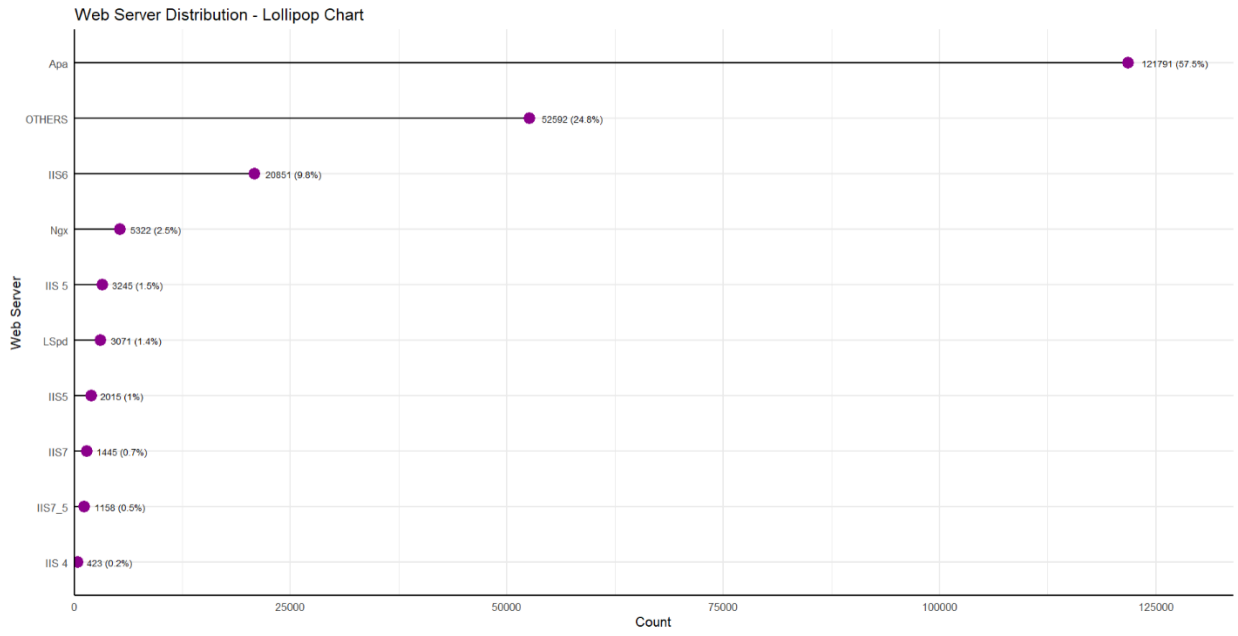


Figure 34: 3.1.1 Analysis 1-1.2 Lollipop Chart (2D)

The lollipop chart visualizes the distribution of web servers based on their occurrence in the dataset. Web server is the version used by the server hosting the defaced website. The x-axis represents different web servers, while the y-axis indicates their frequency. The top 10 most commonly used web servers are displayed in descending order, with less frequent ones grouped under "OTHERS". Each web server is marked with a dark magenta point and labeled with its count and percentage. The highest occurrence is "Apa" with (121791, 57.5%). The y-axis is scaled up to 125,000 for better readability. This chart highlights dominant web servers in the dataset, providing clear insights in distribution and usage patterns.

3.1.1 Analysis 1-1.2: Lollipop Chart: The distribution of Top 5 Webserver based on Count and Percentage (3D Graph)

```
# Create 3D Lollipop Chart
p <- plot_ly() %>%
  add_trace(
    type = "scatter3d",
    mode = "lines+markers",
    x = ~top_5_webservers$WebServer,
    y = ~top_5_webservers$Count,
    z = ~top_5_webservers$Percentage,
    line = list(color = "darkgray", width = 3),  # Connecting line
    marker = list(size = 6, color = "darkmagenta")  # Lollipop head
  ) %>%
  layout(
    title = "Web Server Distribution - 3D Lollipop Chart",
    scene = list(
      xaxis = list(title = "Web Server", tickangle = 45),
      yaxis = list(title = "Count"),
      zaxis = list(title = "Percentage (%)")
    ),
    margin = list(l = 10, r = 10, b = 10, t = 10)
  )

# Print the 3D plot
p
```

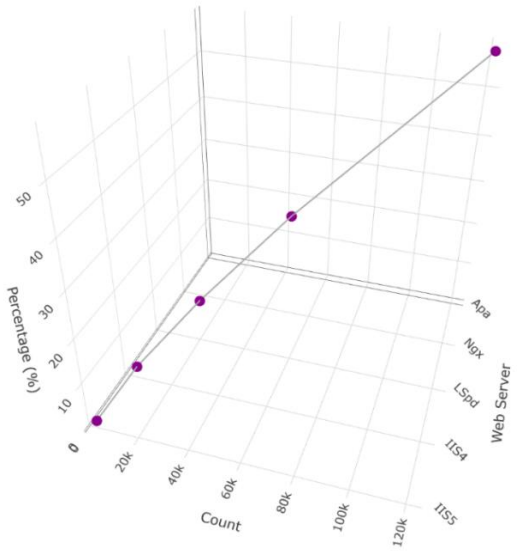


Figure 35: 3.1.1 Analysis 1-1.2: Lollipop Chart Code (3D)

Figure 36: 3.1.1 Analysis 1-1.2: Lollipop Chart (3D)

The 3D lollipop chart visualizes the distribution of web servers based on their occurrence and percentage share in the dataset. Unlike the 2D version, this chart introduces a third dimension, making it easier to compare web servers by count and proportion. The x-axis represents the top five most frequently used web servers, the y-axis shows their occurrence count, and the z-axis indicates their percentage of the total dataset. The ”Apa” (Apache) has the highest count (121791, 57.5%) compared to the rest of the webservers.

Each lollipop consists of a dark gray line extending to a dark magenta marker, highlighting exact values. To maintain clarity, all less common web servers are grouped under "OTHERS". The tilted x-axis labels improve readability, and dynamically selecting the top five web servers ensures the chart remains relevant.

3.1.2 Analysis 1-2: Bar Chart: The distribution of Top 10 Webserver across Ransom Attacks

```
# Create the bar plot
p <- ggplot(final_counts, aes(x = WebServer, y = AttackCount, fill = WebServer)) +
  geom_bar(stat = "identity", alpha = 0.7) +
  scale_fill_manual(values = webserver_colors) +
  labs(title = "Distribution of Ransom Attacks Across Web Servers",
       x = "Web Server (Short Name)",
       y = "Count of Ransom Attacks") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the plot
print(p)
```

Figure 37: 3.1.2 Analysis 1-2: Bar Chart Code

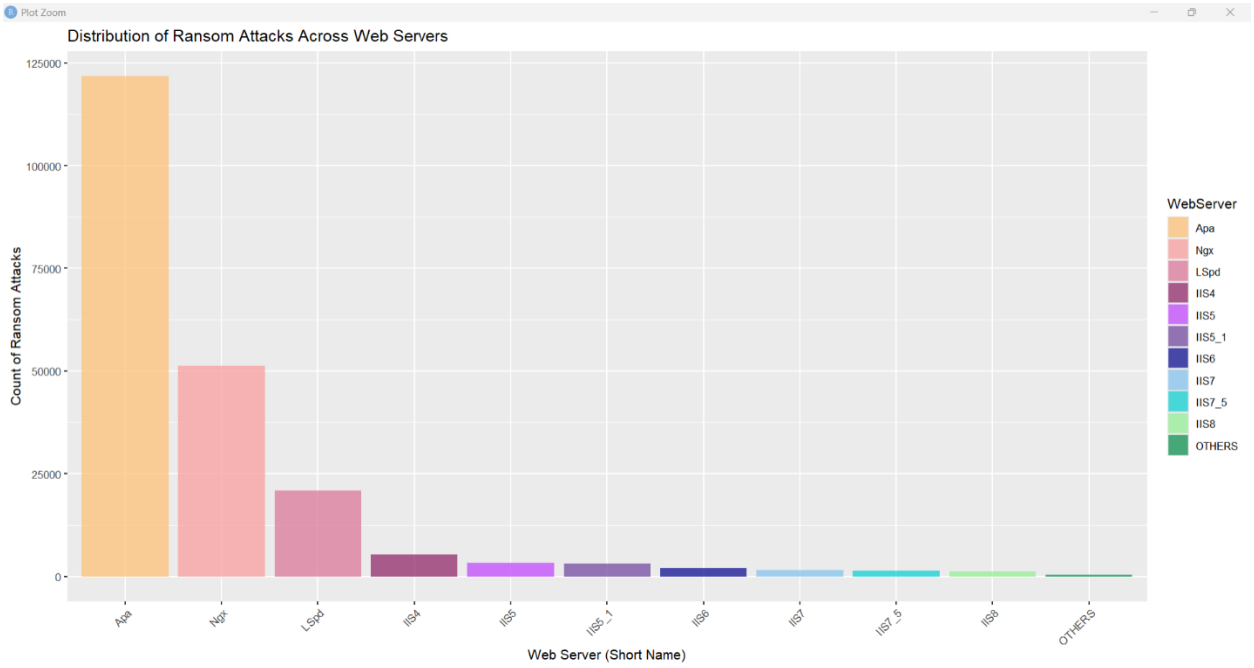


Figure 38: 3.1.2 Analysis 1-2: Bar Chart

The bar chart visualizes the distribution of ransomware attacks across the top 10 web servers in the dataset. Web server is the version used by the server hosting the defaced website whereas ransom is the amount paid in Thousands. The x-axis represents web servers using shortened names

for clarity, while the y-axis indicates the total number of attacks associated with each server. Each web server is assigned a distinct color, and the x-axis labels are tilted to enhance readability. The height of each bar illustrates the frequency of attacks, making it easy to identify the most targeted web servers. Web servers outside the top 10 are grouped under "OTHERS" to maintain focus on the most affected platforms. Furthermore, this graph is displayed in descending order where ”Apa” (Apache) having highest between 100000-125000 ransom attacks whereas ”OTHERS” the lowest which is less than 25000. This visualization effectively highlights security risks, providing insight into which web servers are more vulnerable to ransomware attacks.

3.1.3 Analysis 1-3: Histogram: The distribution of Top 10 Countries across Ransom Attacks

```
# Plot histogram with a suitable scale for y-axis
histogram_plot <- ggplot(top_10_countries, aes(x = reorder(Country, -Total_Downtime), y = Total_Downtime, fill = Country)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 10 Countries with Highest Downtime",
       x = "Country",
       y = "Total Downtime") +
  scale_y_continuous(labels = scales::comma, breaks = seq(0, max(top_10_countries$Total_Downtime), by = 500000)) +  # Custom y-axis breaks
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none")

# Display the plot
print(histogram_plot)
```

Figure 39: 3.1.3 Analysis 1-3: Histogram Code

	Country	Total_Downtime
	<chr>	<dbl>
1	UNITED STATES	3657814
2	UNKNOWN	338226
3	GERMANY	219908
4	BRAZIL	211126
5	SPAIN	199046
6	UNITED KINGDOM	176534
7	FRANCE	164460
8	CHINA	101036
9	CANADA	97873
10	ITALY	96124

Figure 40: 3.1.3 Analysis 1-3: Histogram (Table Output)

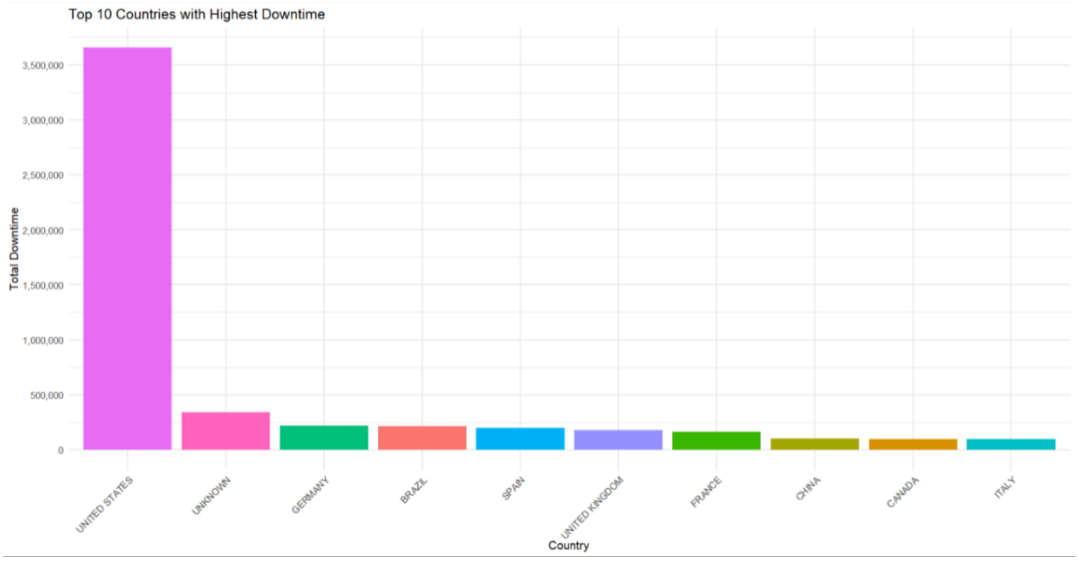


Figure 41: 3.1.3 Analysis 1-3: Histogram Chart

The histogram visualizes the total downtime across the top 10 countries with the highest recorded downtime in the dataset. Country is where the server hosting the defaced website is located whereas downtime is the number of days when a system is unavailable. . The x-axis represents the

countries, while the y-axis shows the total downtime, formatted with commas for clarity. Each country is assigned a distinct color, ensuring easy differentiation. The bars are arranged in descending order, highlighting the countries most affected by downtime. For example, in this top 10 data, Italy is least affected by downtime lesser than 500000. To improve readability, the x-axis labels are tilted. Based on the output table, Italy total downtime is 96124. This visualization effectively identifies regions experiencing the most significant downtime, providing insights into geographical patterns of service disruptions.

3.1.4 Analysis 1-4: Pie Chart: The distribution of Top 5 Countries across Ransom Attacks

```
# Create the pie chart
pie_chart <- ggplot(top_5_countries, aes(x = "", y = TotalRansom, fill = Label)) +
  geom_bar(stat = "identity", width = 1, color = "black") +  # Creates pie chart
  coord_polar(theta = "y", start = 0) +  # Converts to circular pie chart
  scale_fill_manual(values = country_colors) +  # Apply colors
  theme_void() +  # Remove unnecessary background elements
  labs(title = "Top 5 Countries with Ransom Attacks", fill = "Country (Total Ransom)") +
  theme(legend.position = "right")  # Position legend to the right

# Display the pie chart
print(pie_chart)

# Print summary table with Total Ransom Attacks
top_5_countries %>%
  select(Country, TotalRansom) %>%
  kable(caption = "Ransom Attacks per Country (Top 5)")
```

Figure 42: 3.1.4 Analysis 1-4: Pie Chart Code

Country	TotalRansom
El Salvador	180724396
North Korea	18133019
Timor-Leste	12233481
Guam	11927693
Bouvet Island	10405151

Figure 43: 3.1.4 Analysis 1-4: Pie Chart (Table output)

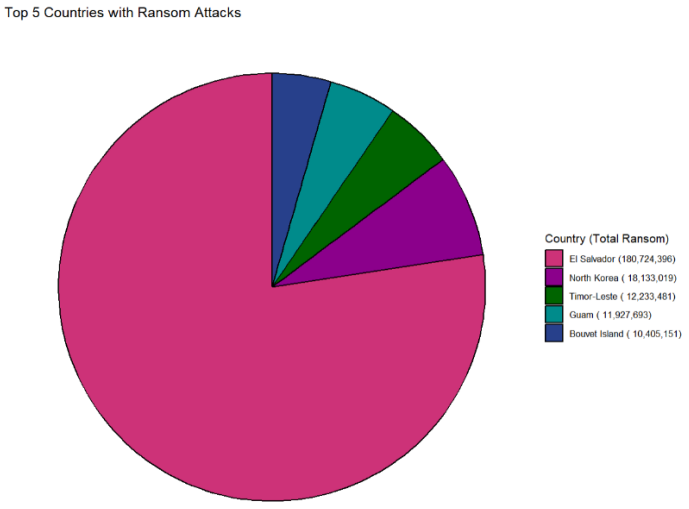


Figure 44: 3.1.4 Analysis 1-4: Pie Chart

The pie chart displays ransomware attacks across the top five affected countries, with segment sizes representing total ransom amounts. Country is where the server hosting the defaced website is located whereas ransom is the amount paid in thousands. Distinct colors differentiate the

countries, and the legend clarifies their proportions. The chart highlights the uneven distribution of ransom paid, revealing significant disparities.

El Salvador, in violet-red, dominates with a ransom total of 180,724,396, far surpassing the others. Its large segment emphasizes its high vulnerability, suggesting it is a prime target for cybercriminals. This stark contrast underscores the urgent need for stronger cybersecurity.

North Korea, Timor-Leste, Guam, and Bouvet Island have much smaller segments, indicating lower attack frequencies. The color contrast effectively shows which nations are most at risk, providing a clear visual comparison.

3.1.5 Analysis 1-5: Bar Chart with Line Graph: Web Servers and Ransom Attacks based on Downtime

```
# Create a stacked bar chart with a line graph and red points
stacked_line_chart <- ggplot(downtime_summary, aes(x = reorder(WebServer, -TotalDowntime))) +
  geom_bar(aes(y = TotalDowntime, fill = factor(TotalRansomAttacks)), stat = "identity") +
  geom_line(aes(y = TotalDowntime, group = 1), color = "darkslateblue", size = 1) +
  geom_point(aes(y = TotalDowntime), color = "red", size = 1.5) +  # Add red dots
  scale_y_continuous(limits = c(0, y_max), breaks = y_breaks) +
  labs(title = "Web Servers and Ransom Attacks Based on Downtime",
       x = "Web Server",
       y = "Total Downtime",
       fill = "Number of Ransom Attacks",
       color = "Downtime") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the stacked bar chart with red points on the line graph
print(stacked_line_chart)
```

Figure 45: 3.1.5 Analysis 1-5: Bar Chart with Line Graph Code

	WebServer	TotalDowntime	TotalRansomAttacks	DowntimePercentage	RansomPercentage
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Apa	3879480	188150670.	57.4	57.4
2	Ngx	1637453	79354782.	24.2	24.2
3	LSpd	665649	32232990.	9.9	9.8
4	IIS4	168425	8188312.	2.5	2.5
5	IIS5	103567	5014284.	1.5	1.5
6	IIS5_1	96891	4747827.	1.4	1.4
7	IIS6	64102	3116560.	0.9	1
8	IIS7	45594	2239444.	0.7	0.7
9	IIS7_5	42182	2031775.	0.6	0.6
10	IIS8	37508	1799157.	0.6	0.5
11	OTHERS	12946	666096.	0.2	0.2

Figure 46: 3.1.5 Analysis 1-5 (Table Output)

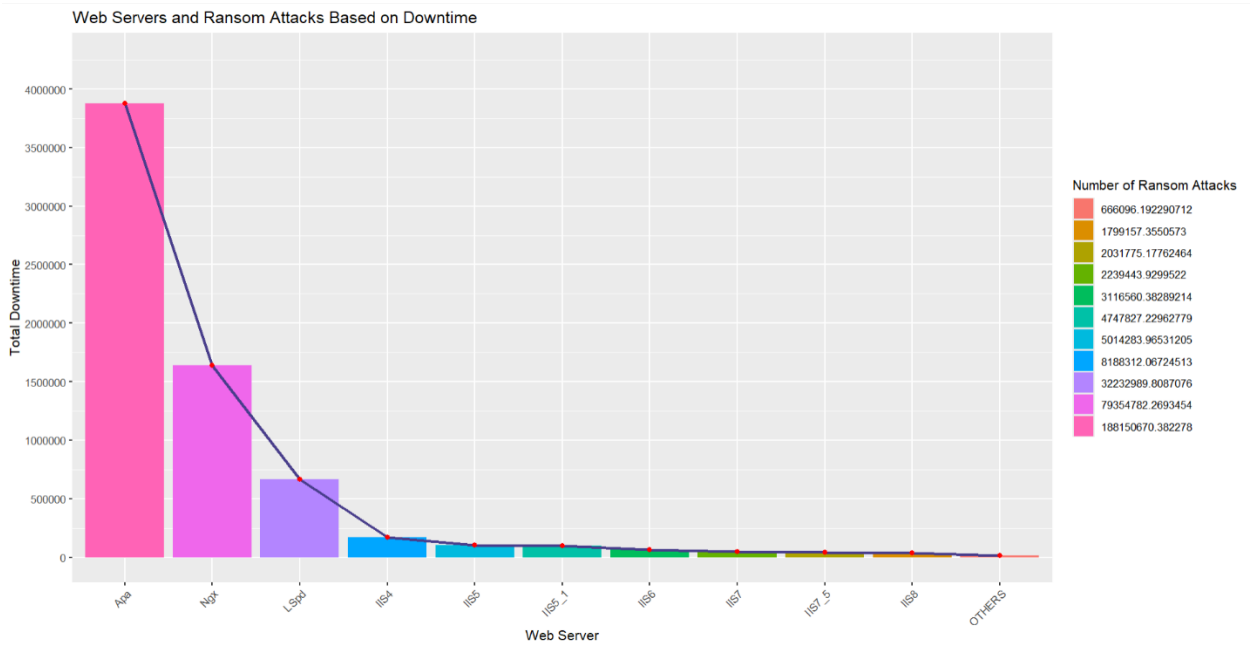


Figure 47: 3.1.5 Analysis 1-5: Bar Chart with Line Graph

The graph illustrates the relationship between web servers, total downtime, and the number of ransomware attacks. The height of each bar represents the total downtime in days unavailable system experienced by each web server used by server hosting that defaced website, while the colors indicate the amount of ransom attacked paid. The overlaid line graph, marked with red points, tracks downtime trends across different web servers, providing a visual representation of the impact of cyber incidents.

Apache, represented by the tallest pink bar, has the highest total downtime, exceeding 4,000,000, making it the most affected web server leads to significantly prolonged outages, which could indicate its widespread use or vulnerabilities that cybercriminals exploit. The "OTHER" category is less downtime, suggesting a collective group of various web servers experiencing similar attacks but at a lower magnitude.

As the graph progresses, downtime decreases sharply across less affected servers, including LSpd, IIS4, and others. These servers have notably lower downtime, reflected in smaller bars and a flatter line graph. The color variation indicates differences in ransomware attack frequency, with the legend on the right providing insight into attack numbers. Overall, chart highlights specific web

servers experience disproportionate impacts from ransomware, emphasizing the need for improved security measures in highly targeted systems like Apache.

3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph: Web Server, Ransom Attacks, Downtime, and Country.

```
# Create a clustered bar chart with line graph (red dot for Ransom)
clustered_bar_line_chart <- ggplot(downtime_summary, aes(x = reorder(WebServer, -TotalDowntime), y = TotalDowntime, fill = WebServer)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7)) +  # Bars for Downtime
  geom_line(aes(y = TotalRansom * scale_factor, group = 1), color = "black", size = 1) +  # Red Line for Ransom
  geom_point(aes(y = TotalRansom * scale_factor), color = "red", size = 3) +  # Red Dots for Ransom
  geom_text(aes(x = WebServer, y = TotalRansom * scale_factor, label = TotalRansom),
            color = "black",
            vjust = ifelse(downtime_summary$Index == 1, -1.4, -1.8),  # Keep first label at -1.4, adjust others
            angle = ifelse(downtime_summary$Index == 1, 0, 8)) +  # Keep first label at angle 0, others at 8
  scale_fill_manual(values = setNames(downtime_summary$Color, downtime_summary$WebServer)) +  # Apply custom colors
  scale_y_continuous(name = "Total Downtime") +
  labs(title = "Clustered Bar Chart of Downtime by WebServer (USA) with Ransom Trends",
       x = "Web Server",
       fill = "Web Server",
       caption = "Note: Red line & dots represent ransom attacks (scaled for visualization)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Print the final graph
print(clustered_bar_line_chart)
```

Figure 48: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph Code

Show 5 entries Search:

	WebServer	TotalDowntime	TotalRansom
1	Apa	2443065	118626181.2062275
2	LSpd	495707	23940459.21094969
3	OTHER	463144	22506141.99825316
4	IIS4	65399	3222508.528919437
5	IIS5_	57590	2812202.484537275

Figure 49: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 1)

Show 5 entries Search:

	WebServer	TotalDowntime	TotalRansom
6	IIS5	49057	2369569.600944468
7	IIS6	38594	1886022.295913082
8	IIS7	25797	1259561.098074599
9	IIS7_	15902	756781.7228034489
10	IIS8	3559	198375.7467797979

Figure 50: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 2)

```
Table: Summary of WebServer, Ransom, and Downtime (Top 10 in the United States)

|WebServer | TotalDowntime| TotalRansom|
|:---------|-------------:|-----------:|
|Apa       |       2443065| 118626181.2|
|LSpd      |        495707|  23940459.2|
|OTHER     |        463144|  22506142.0|
|IIS4      |         65399|   3222508.5|
|IIS5_     |         57590|   2812202.5|
|IIS5      |         49057|   2369569.6|
|IIS6      |         38594|   1886022.3|
|IIS7      |         25797|   1259561.1|
|IIS7_     |         15902|    756781.7|
|IIS8      |          3559|    198375.7|
```

Figure 51: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Table Output 3)

```
> total_rows <- nrow(us_data)
> print(paste("Total number of rows used in the analysis:", total_rows))
[1] "Total number of rows used in the analysis: 114963"
```

Figure 52: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph (Print Total Rows)

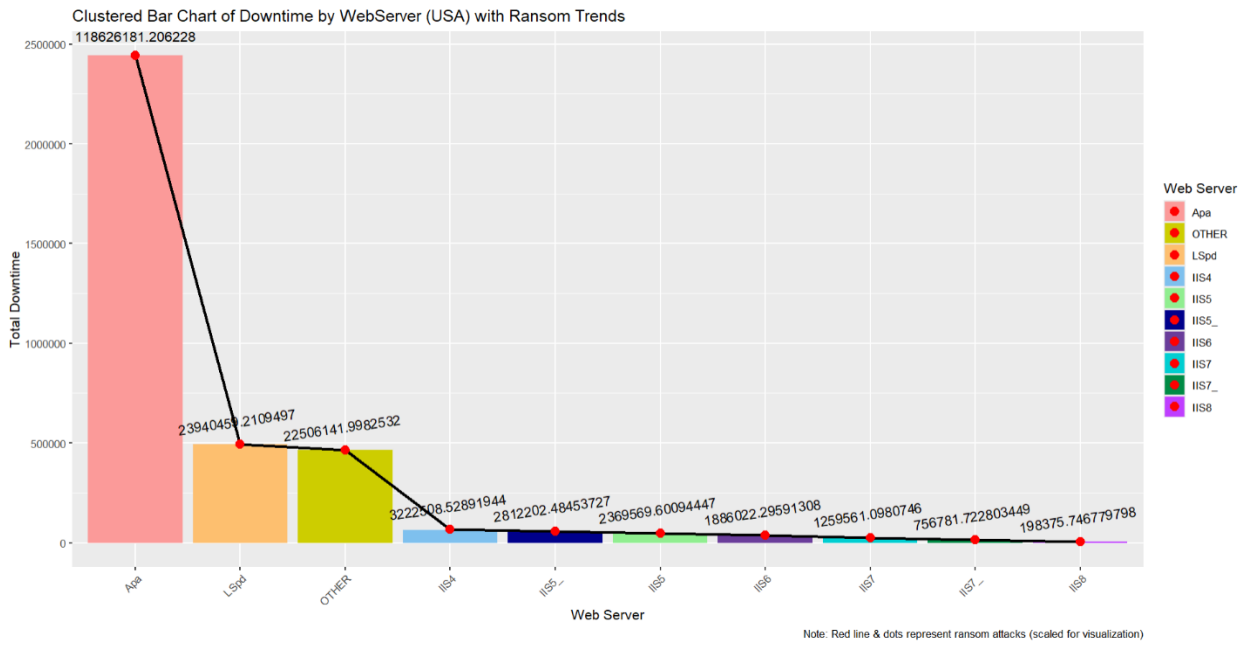


Figure 53: 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph

This clustered bar chart shows the total unavailable system in days of downtime with various web servers used by server hosting in the United States alongside ransom attack trends paid. The x-axis represents web servers, while the y-axis quantifies downtime. Each colored bar indicates downtime for each web server, and the red line with dots represents total ransom paid, scaled for visualization.

The data reveals that Apache (Apa) experiences the highest downtime, exceeding 11.86 million, far surpassing other servers. LiteSpeed (LSpd) and "OTHER" servers follow with significantly lower downtime, around 2.39 million and 2.25 million, respectively. Other web servers, including IIS4, IIS5, IIS6, IIS7, and IIS8, show minimal downtime, suggesting lower impact.

Ransom attack trends, shown by the red dots and black line, decrease as downtime reduces. Apache has the highest ransom attacks, while IIS servers show relatively low ransom figures, indicating fewer ransomware disruptions.

The graph effectively combines bar and line charts to explore the relationship between downtime and ransom attacks. The note clarifies that ransom values are scaled for better representation, and the dataset ensures a thorough analysis of downtime and security risks among the top U.S. web servers.

Conclusion based on 3.1.6 Analysis 1-6: Clustered Bar Chart with Line Graph: Web Server, Ransom Attacks, Downtime, and Country.

```
> # ---------------- Chi-Square Test ---------------- #
> # Checking if WebServer distribution depends on Country
> webserver_country_table <- table(data$WebServer, data$Country)
>
> if (any(webserver_country_table < 5)) {
+   chi_sq_test <- if (sum(webserver_country_table) < 200) {
+     fisher.test(webserver_country_table)  # Fisher's Exact Test for small tables
+   } else {
+     chisq.test(webserver_country_table, simulate.p.value = TRUE)  # Simulate p-value if necessary
+   }
+ } else {
+   chi_sq_test <- chisq.test(webserver_country_table)  # Standard Chi-Square Test
+ }
> print(chi_sq_test)

        Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data:  webserver_country_table
X-squared = 354140, df = NA, p-value = 0.0004998
```

Figure 54: Conclusion (Chi-Square Test)

```
> # ---------------- T-Test / ANOVA ---------------- #
> # Checking normality before deciding test type (limit to 5000 samples for efficiency)
> sample_size <- min(5000, nrow(data))
>
> shapiro_ransom <- shapiro.test(sample(data$Ransom, sample_size))$p.value
> shapiro_downtime <- shapiro.test(sample(data$Downtime, sample_size))$p.value
>
> # Define appropriate statistical test based on normality
> ransom_test <- if (shapiro_ransom > 0.05) {
+   aov(Ransom ~ WebServer, data = data)
+ } else {
+   kruskal.test(Ransom ~ WebServer, data = data)
+ }
> print(summary(ransom_test))
           Length Class  Mode
statistic  1      -none- numeric
parameter  1      -none- numeric
p.value    1      -none- numeric
method     1      -none- character
data.name  1      -none- character
>
> downtime_test <- if (shapiro_downtime > 0.05) {
+   aov(Downtime ~ WebServer, data = data)
+ } else {
+   kruskal.test(Downtime ~ WebServer, data = data)
+ }
> print(summary(downtime_test))
           Length Class  Mode
statistic  1      -none- numeric
parameter  1      -none- numeric
p.value    1      -none- numeric
method     1      -none- character
data.name  1      -none- character
```

Figure 55: Conclusion (T-Test)

```
> # ---------------- Correlation Test ---------------- #
> # Remove missing values for accurate correlation analysis
> filtered_data <- data[!is.na(Ransom) & !is.na(Downtime)]
>
> # Downsample the data (e.g., take 5000 random rows for testing)
> set.seed(42)  # Set seed for reproducibility
> sample_size <- min(5000, nrow(filtered_data))  # Limit to 5000 rows
> filtered_data_sample <- filtered_data[sample(1:nrow(filtered_data), sample_size), ]
>
> # Use Kendall's correlation for non-parametric data
> correlation_test <- cor.test(filtered_data_sample$Ransom, filtered_data_sample$Downtime, method = "kendal
l")
> print(correlation_test)

        Kendall's rank correlation tau

data:  filtered_data_sample$Ransom and filtered_data_sample$Downtime
z = 0.52692, p-value = 0.5982
alternative hypothesis: true tau is not equal to 0
sample estimates:
        tau
0.005779653
```

Figure 56: Conclusion (Correlation Test)

The Chi-Square Test assesses the relationship between web server types and countries regarding ransom attacks. The test returned a p-value of 0.0004998, which is below the significance level of 0.05, suggesting a significant relationship between these variables. This indicates that certain countries and web servers are more likely to experience ransom attacks than others.

For the T-Test/ANOVA, the differences in ransom and downtime across web servers were tested. Due to non-normality in the data (as indicated by the Shapiro-Wilk test for normality), non-parametric tests like Kruskal-Wallis were applied. However, the summary of the tests did not provide specific results. If the p-values from these tests are below 0.05, it would suggest significant differences in ransom and downtime between web servers.

The Kendall's Rank Correlation test assesses the relationship between ransom attack frequency and downtime. The result showed a p-value of 0.5982, which is much higher than 0.05, indicating no significant correlation between the frequency of ransom attacks and downtime. This suggests that the number of attacks does not significantly impact the length of downtime.

In conclusion, the Chi-Square test supports the hypothesis that certain web servers and countries are more vulnerable to attacks. However, the T-Test and Kendall's Rank Correlation tests suggest no significant differences or correlations in downtime based on the type of web server or the frequency of ransom attacks.

Additional Feature

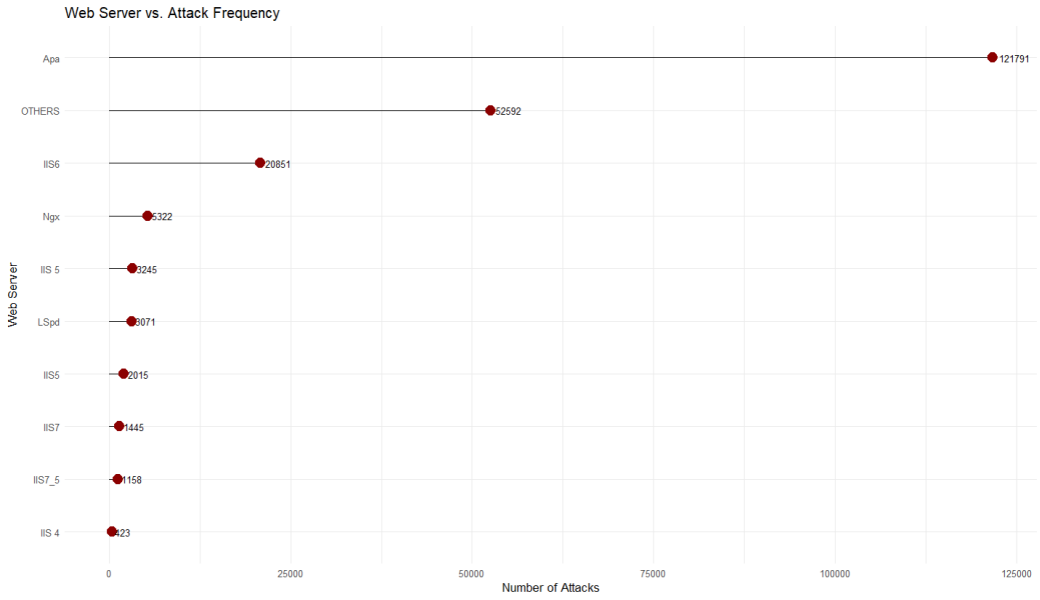
Data Cleaning and Preprocessing	1. Handling Missing Values • is.na(), na.rm = TRUE: Manage missing data. • Replacing empty strings with "Unknown". 2. Renaming & Shortening Web Server Names • sapply() with short_names dictionary: Standardizes web server names. 3. Grouping Data into "OTHERS" • ifelse(): Classifies less frequent web servers under "OTHERS". • Merging "N/A" values into "OTHERS" for clarity. 4. Sorting Data Dynamically • top_n(), arrange(desc(Count)): Selects and sorts the top web servers/countries. 5. Data Type Conversion & Normalization • as.numeric(), as.factor(), toupper(): Converts data types and ensures consistency.
Data Aggregation & Statistical Analysis Features	1. Summarizing Data • group_by(), summarise(): Calculate counts, ransom attacks, and total downtime. • mutate(Percentage = Count / sum(Count) * 100): Computes percentage distributions. 2. Chi-Square Test for Categorical Data • chisq.test() and fisher.test(): Analyzes relationships between categorical variables. 3. T-Test & ANOVA for Mean Comparison • aov(), kruskal.test(): Compares ransom amounts and downtime across web servers. • shapiro.test(): Checks normality to decide between parametric and non-parametric tests. 4. Correlation Analysis: • cor.test(method = "kendall"): Measures the relationship between ransom frequency and downtime. 5. Contingency Tables for Chi-Square Analysis: • table(): Creates frequency tables for categorical analysis.
Data Visualization Features	1. Custom Color Schemes • scale_fill_manual(values = c(...)): Applies specific colors to charts. 2. Lollipop Chart for Web Server Distribution • geom_segment(), geom_point(): Creates a lollipop chart. 3. Bar Chart & Line Graph Combination • geom_bar(), geom_line(), geom_point(): Combines different visual elements. 4. Interactive 3D Plot with Plotly • plot_ly(), add_trace(type = "scatter3d"): Builds 3D visualizations. 5. Circular (Polar) Charts

	• coord_polar(): Converts bar charts to pie/donut charts. 6. Faceting for Multi-Panel Plots • facet_wrap(~ variable): Splits data into multiple subplots. 7. Text Annotations & Labels • geom_text(), paste0(): Adds labels to charts. 8. Theme Customization • theme_minimal(), axis.text.x = element_text(angle = 45, hjust = 1): Enhances chart appearance
Table Display & Data Output Features	1. Creating & Displaying Tables: • kable(): Creates static summary tables. • datatable(): Produces interactive tables with sorting and searching capabilities.

3.2 Objective 2: The study analyzes web server resilience to ransom attacks by examining downtime, attack frequency, and notification impact across countries (Lim Eazen TP078935)

3.2.1 Which web servers are most frequently attacked, and what does this reveal about their resilience against cyber threats?

```
#1. Web Server & Attack Frequency (Lollipop Chart)

p1 <- ggplot(webserver_counts, aes(x = reorder(WebServer, Count), y = Count)) +
  geom_segment(aes(xend = WebServer, yend = 0), color = "black") +
  geom_point(color = "darkred", size = 4) +
  geom_text(aes(label = Count), hjust = -0.2, size = 3, check_overlap = TRUE) +
  coord_flip() +
  labs(title = "Web Server vs. Attack Frequency", x = "Web Server", y = "Number of Attacks") +
  theme_minimal()
print(p1)
```



The lollipop chart visualizes attack frequency on web servers, with Apache experiencing the most attacks, exceeding 120,000. "OTHERS" also shows high attack counts, indicating frequent targeting of multiple lesser-known servers. IIS6, Nginx, and IIS5 follow but at lower rates.

However, attack frequency alone does not indicate resilience. A highly attacked server may recover quickly, while a less attacked one may suffer prolonged downtime. To assess resilience, integrating downtime and recovery data is essential. A scatter plot comparing attack frequency vs.

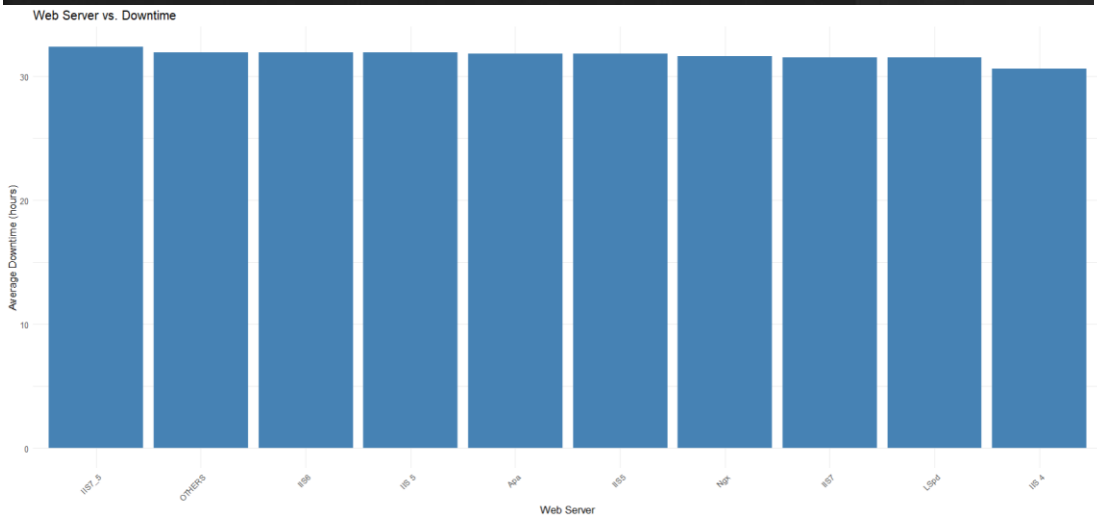
downtime or a stacked bar chart showing average downtime per attack would provide deeper insights.

Conclusion:

Apache is the most frequently attacked server, but attack frequency alone does not indicate resilience.

3.2.2 Which web servers experience the longest downtime, and what does this indicate about their resilience in recovering from attacks?

```r
#2. Web Server & Downtime (Boxplot)
downtime_data <- data %>%
  group_by(WebServer) %>%
  summarise(Average_Downtime = mean(Downtime, na.rm = TRUE)) %>%
  arrange(desc(Average_Downtime))

p2 <- ggplot(downtime_data, aes(x = reorder(WebServer, -Average_Downtime), y = Average_Downtime)) +
  geom_col(fill = "steelblue") +  # Use bar chart instead
  labs(title = "Web Server vs. Downtime", x = "Web Server", y = "Average Downtime (hours)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(p2)
```



The bar chart measures web server resilience by analyzing downtime. While attack frequency shows which servers are targeted, downtime reveals recovery efficiency. Servers with high attack rates but quick recovery are more resilient.

The code groups data by server, calculates average downtime (ignoring missing values), and sorts results in descending order. The bar chart (geom_col) visually ranks servers, with reordered x-axis labels for clarity.

Results show nearly identical downtime across servers, suggesting recovery depends on external factors rather than server type. Apache and IIS6, despite frequent attacks, recover as quickly as less-targeted servers, indicating strong resilience. This may reflect industry-wide recovery standards.

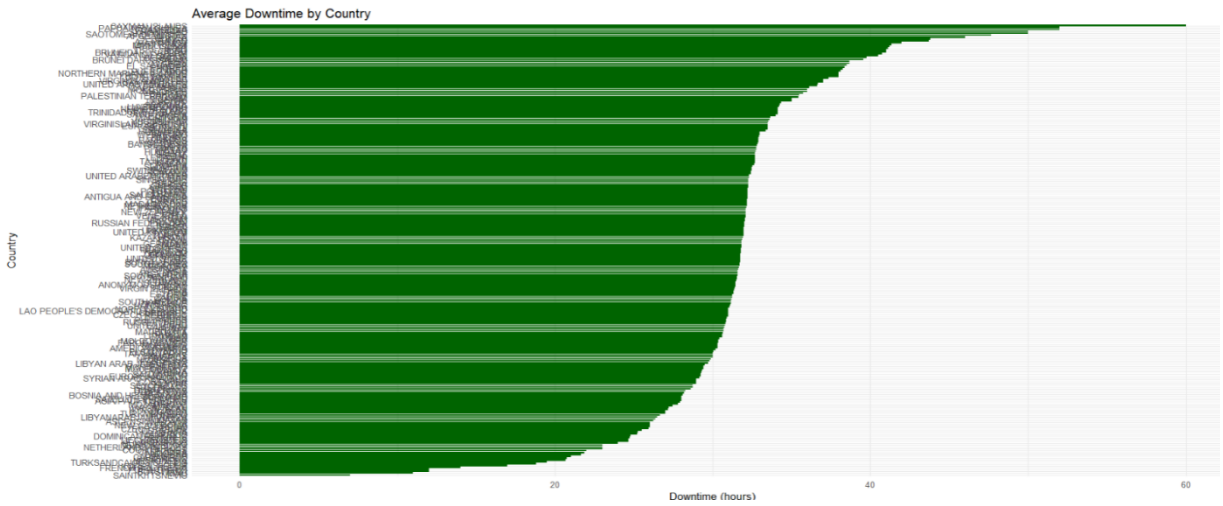
Comparing attack frequency with downtime emphasizes that resilience depends on recovery speed, not just prevention.

Conclusion:

Servers with high attack frequency like Apache maintain similar downtime to other servers, indicating efficient recovery mechanisms.

3.2.3 Which countries experience the longest downtime, and how does this reflect their resilience in mitigating cyberattacks?

```
#3. Downtime & Country (Bar Chart)

country_downtime <- data %>%
  group_by(Country) %>%
  summarise(Average_Downtime = mean(Downtime, na.rm = TRUE)) %>%
  arrange(desc(Average_Downtime))

p3 <- ggplot(country_downtime, aes(x = reorder(Country, Average_Downtime), y = Average_Downtime)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  coord_flip() +
  labs(title = "Average Downtime by Country", x = "Country", y = "Downtime (hours)") +
  theme_minimal()
print(p3)
```

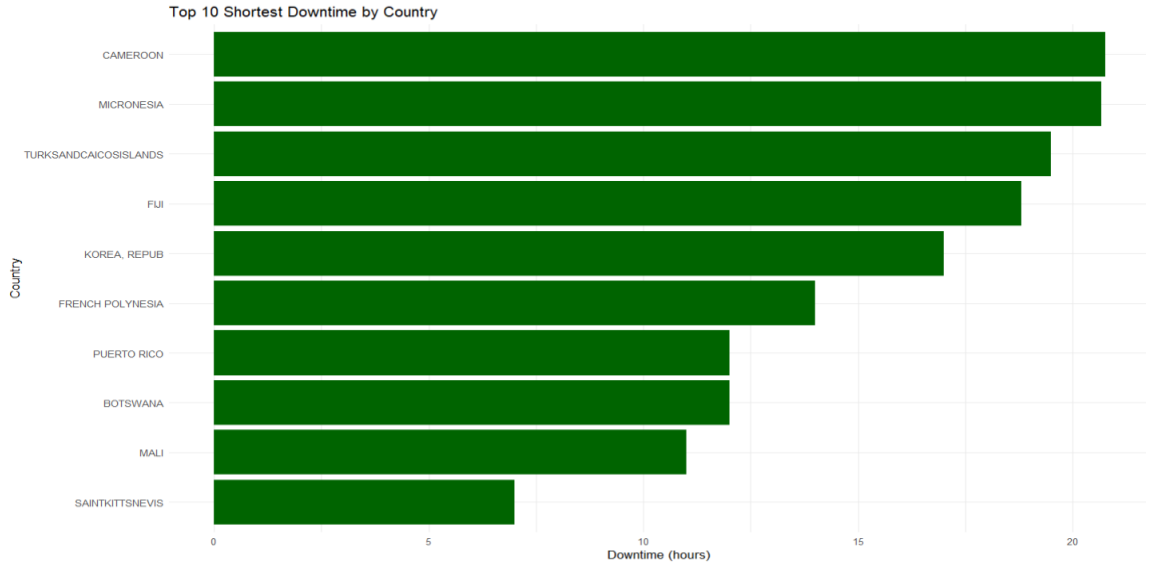


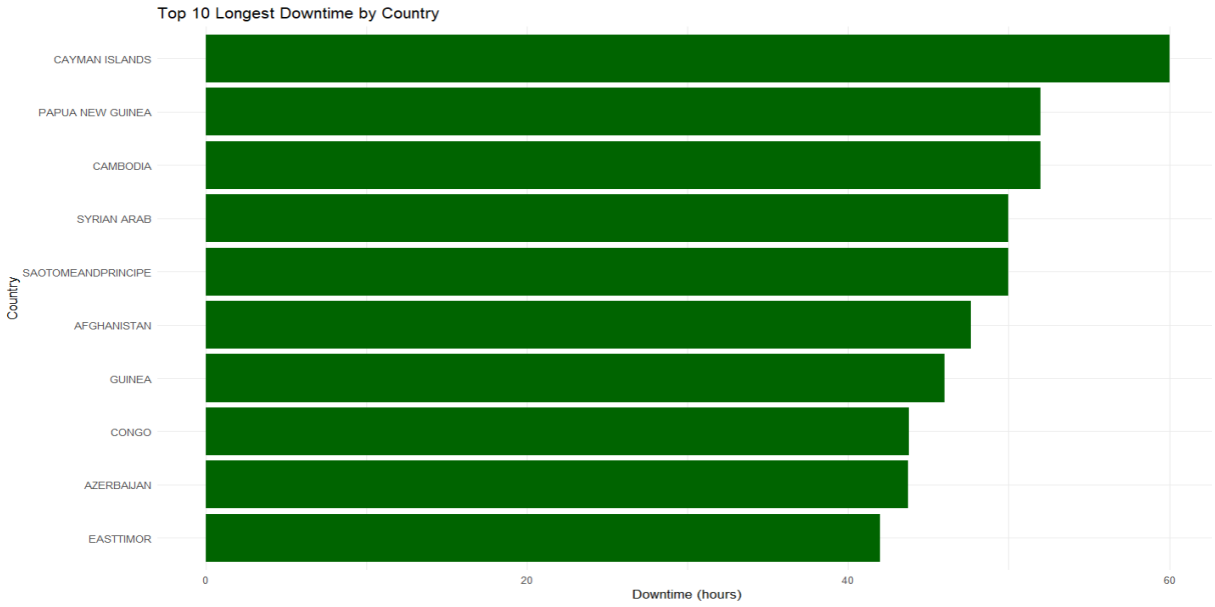
The code calculates average downtime per country by grouping data, computing the mean (excluding missing values), and sorting results in descending order. The horizontal bar chart (p3) ranks countries by downtime, offering a clear comparison.

To refine the analysis, it selects the 10 countries with the longest (top_longest) and shortest (top_shortest) downtimes. The visualizations (p3_1, p3_2) highlight regions struggling with downtime and those maintaining stable web services.

This analysis reveals geographical trends in server downtime, influenced by infrastructure, cybersecurity, and response efficiency. Regions with high downtime may require better maintenance and recovery strategies, while those with low downtime likely have strong resilience mechanisms.

```
# Select top 5 longest and 5 shortest downtime countries
top_longest <- country_downtime %>% top_n(10, Average_Downtime)
top_shortest <- country_downtime %>% top_n(-10, Average_Downtime)


# Plot
p3_1 <- ggplot(top_longest, aes(x = reorder(Country, Average_Downtime), y = Average_Downtime)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  coord_flip() +
  labs(title = "Top 10 Longest Downtime by Country", x = "Country", y = "Downtime (hours)") +
  theme_minimal()

print(p3_1)

p3_2 <- ggplot(top_shortest, aes(x = reorder(Country, Average_Downtime), y = Average_Downtime)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  coord_flip() +
  labs(title = "Top 10 Shortest Downtime by Country", x = "Country", y = "Downtime (hours)") +
  theme_minimal()

print(p3_2)
```





The data shows that countries with the longest downtimes, such as Cayman Islands, Papua New Guinea, and Cambodia, often have fragile infrastructure, unstable internet, or political instability. Conflict zones like Syria and Afghanistan also face extended downtimes due to disrupted services.

Conversely, countries with the shortest downtimes, like Cameroon, Micronesia, and Fiji, likely benefit from stable infrastructure or efficient recovery. Advanced nations like South Korea showcase strong network resilience, while Botswana and Mali's presence suggests some

developing nations have achieved reliable systems through infrastructure investments or telecom partnerships.

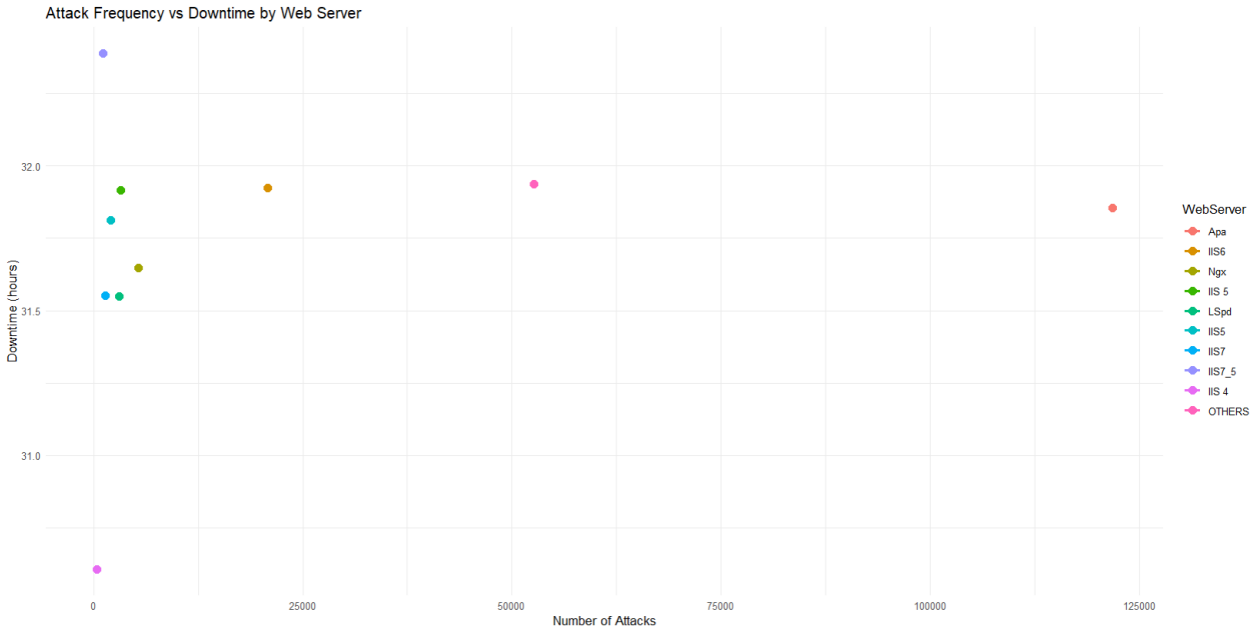
Geographical location, economic stability, and policies influence downtime management. High-downtime countries could improve reliability with better infrastructure, disaster recovery plans, and stronger telecom policies.

Conclusion:

Countries with fragile infrastructure, such as island nations or conflict zones, tend to experience longer downtime.

3.2.4 How does the frequency of attacks on different web servers correlate with their downtime, and which servers demonstrate the strongest resilience against frequent cyberattacks?

```r
#4. Web Server, Attack Frequency, and Downtime (Scatter Plot)

server_attack_downtime <- data %>%
  group_by(WebServer) %>%
  summarise(Attack_Frequency = n(),
            Average_Downtime = mean(Downtime, na.rm = TRUE)) %>%
  arrange(desc(Attack_Frequency))


p4 <- ggplot(server_attack_downtime, aes(x = Attack_Frequency, y = Average_Downtime, color = WebServer)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Attack Frequency vs Downtime by Web Server", x = "Number of Attacks", y = "Downtime (hours)") +
  theme_minimal()
print(p4)
```



This analysis evaluates web server resilience by examining attack frequency and downtime. By grouping servers and calculating both metrics, we identify which servers withstand cyber threats effectively. A scatter plot with a trend line highlights correlations, showing whether frequent attacks impact downtime.

Apache (Apa) faces the highest attack rate (100,000+ attacks) yet maintains a downtime of around 32 hours, indicating strong recovery mechanisms. Other servers like IIS6, Nginx (Ngx), and Litespeed (LSpd) experience fewer attacks but similar downtimes, proving that attack frequency alone does not dictate resilience. Factors like server architecture and response efficiency play key roles.

Anomalies, such as IIS4 with low attack frequency and minimal downtime, suggest that some servers are either inherently secure or less targeted. The lack of a clear correlation further emphasizes that recovery efficiency varies regardless of attack volume.

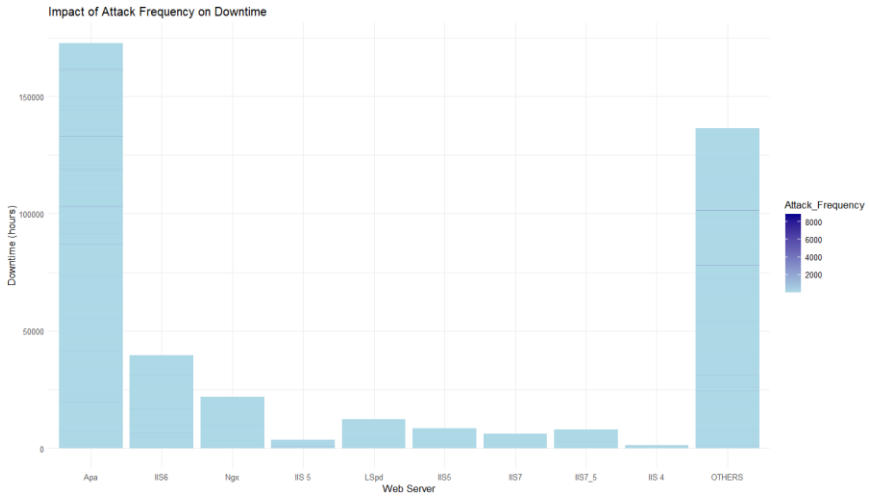
This insight aids in selecting resilient web servers, improving cybersecurity strategies, and enhancing recovery measures to ensure minimal downtime and continuous operations.

Conclusion:

Some web servers like Apache, despite frequent attacks, show strong resilience with minimal downtime.

3.2.5 How does the frequency of ransom attacks across different countries impact the downtime of web servers, and which countries have the fastest recovery times?

```
#5. Web Server, Attack Frequency, Downtime, and Notification Methods (Faceted Bar Chart)

p5_data <- data %>%
  group_by(WebServer, Notify) %>%
  summarise(
    Attack_Frequency = n(),
    Average_Downtime = mean(Downtime, na.rm = TRUE),
    .groups = "drop" # Ensure no unwanted grouping
  )


# Ensure there are no missing values in Average_Downtime
p5_data <- p5_data %>% drop_na(Average_Downtime)

# Create the bar chart
p5 <- ggplot(p5_data, aes(x = WebServer, y = Average_Downtime, fill = Attack_Frequency)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Color scale for Attack Frequency
  labs(title = "Impact of Attack Frequency on Downtime", x = "Web Server", y = "Downtime (hours)") +
  theme_minimal()

print(p5)
```



The provided code contributes to assessing the resilience of web servers against ransom attacks by visualizing the relationship between attack frequency and server downtime. The code uses a bar chart to plot the average downtime of different web servers, with the color gradient representing the frequency of attacks on each server. By analyzing this chart, we can identify which servers experience more downtime and how frequently attacks contribute to this downtime.

This visualization is key to achieving the objective, as it directly correlates the frequency of attacks with the downtime experienced by servers. By examining the results, we can identify which servers are most vulnerable to frequent attacks, highlighting those that may need better resilience strategies. Additionally, the chart helps pinpoint which countries face the most downtime, providing insight into the broader regional patterns of server resilience in response to ransom attacks. Ultimately, this analysis aids in understanding how different web servers perform under varying levels of attack pressure and their recovery times, which is critical for improving cybersecurity measures.

```
calculate_total_downtime <- function(data) {
  total_downtime <- data %>%
    group_by(WebServer) %>%
    summarise(Total_Average_Downtime = sum(Average_Downtime, na.rm = TRUE))

  return(total_downtime)
}

total_downtime_result <- calculate_total_downtime(notification_data)
print(total_downtime_result)
```

	WebServer	Total_Average_Downtime
	<fct>	<dbl>
1	Apa	172802.
2	IIS6	39688.
3	Ngx	21791.
4	IIS 5	3536.
5	LSpd	12491.
6	IIS5	8661.
7	IIS7	6307.
8	IIS7_5	8079.
9	IIS 4	1440.
10	OTHERS	136462.

Upon analyzing the data, it was observed that attack frequency is relatively uniform across different servers, as shown by the consistent light blue color on the bar chart. However, downtime varies significantly, revealing differences in resilience.

The downtime statistics for each web server are as follows:

- Apa: 172,802 hours – highest downtime, indicating significant vulnerability.
- IIS6: 39,688 hours – moderate resilience but still high downtime.
- Ngx: 21,791 hours – relatively better performance but still substantial downtime.
- IIS5: 3,536 hours – much lower downtime, showing improved resilience.
- LSpd: 12,491 hours – noteworthy downtime, though not as severe as Apa or IIS6.
- IIS5: 8,661 hours – moderate downtime, lower than IIS6 and LSpd.
- IIS7: 6,307 hours – good resilience and quick recovery.
- IIS7_5: 8,079 hours – relatively low downtime.
- IIS4: 1,440 hours – strong resilience and fast recovery.
- OTHERS: 136,462 hours – a high total downtime, highlighting vulnerabilities.

Despite similar attack rates, downtime differs widely. Servers like Apa and OTHERS have prolonged downtime, suggesting weak recovery strategies. Meanwhile, IIS4 and IIS7 have significantly shorter downtimes, showing strong resilience.

This analysis confirms that attack frequency does not directly correlate with downtime. Some servers recover quickly, while others suffer from extended outages. Identifying these patterns helps improve security measures and recovery pprocesses.

Conclusion:

Apache, despite frequent attacks, maintains recovery times similar to other servers, indicating well-developed resilience mechanisms.

Conclusion

This analysis enhances the understanding of web server resilience and recovery in response to cyberattacks, particularly ransom attacks. The visualizations and statistical analyses provide valuable insights into the relationship between attack frequency, downtime, and server resilience. These findings support your objective of evaluating web server performance under cyber threats and identifying areas for cybersecurity improvement.

Key Findings

- Web Servers & Attack Frequency:
    - Attack frequency does not always correlate with downtime.
    - Apache (Apa) is frequently targeted but recovers faster than some less-attacked servers like IIS5.
    - Recovery time depends on server architecture, response efficiency, and security protocols.
- Downtime & Resilience:
    - Apache (Apa) and IIS6 experience prolonged outages, suggesting recovery gaps.
    - IIS7 and IIS4 show lower downtime, indicating better resilience and faster recovery.
    - The duration of downtime is a crucial factor in assessing resilience.
- Countries & Regional Impact:

  - Developing nations with fragile infrastructure (e.g., Cayman Islands, Papua New Guinea, Cambodia) experience longer downtimes due to unstable internet, power issues, or political instability.
  - Countries with robust infrastructure (e.g., South Korea) recover faster, highlighting the role of advanced mitigation strategies.
- Impact of Ransom Attacks:
  - Countries with weaker infrastructure face longer downtimes due to slower recovery mechanisms.
  - Stronger technological infrastructure enables rapid recovery, showing the importance of efficient recovery protocols.

Final Insights

- Attack frequency alone is not a definitive measure of resilience—recovery speed matters more.
- Servers that experience frequent attacks but recover quickly (e.g., Apache) demonstrate strong resilience.
- Cyber resilience is influenced by both server architecture and regional infrastructure maturity.
- To minimize downtime, investment in server hardening, disaster recovery strategies, and infrastructure improvements is essential.

These findings help in making informed decisions on server selection, cybersecurity investments, and resilience strategies to enhance protection against cyber threats.

Additional Features
Data Cleaning & Preprocessing

• is.na(), na.omit(): Handles missing values.

• gsub(), sub(): Standardizes text data (e.g., renaming web servers).

• ifelse(): Groups rare categories into "OTHERS".

• toupper(), tolower(): Normalizes text case.

Data Aggregation & Statistical Analysis

• group_by(), summarise(): Aggregates data for statistical summaries.

• mutate(): Creates new calculated columns.

• chisq.test(), fisher.test(): Performs Chi-Square test for categorical relationships.

• aov(), kruskal.test(): Compares means using ANOVA or non-parametric tests.

• shapiro.test(): Checks normality of data.

• cor.test(method = "kendall"): Computes correlation between ransom frequency and downtime.

• table(): Creates contingency tables for categorical analysis.

Data Visualization

• ggplot2 functions (geom_bar(), geom_line(), geom_point()): Creates bar charts, line graphs, and scatter plots.

• coord_polar(): Converts bar charts to circular/polar plots.

• facet_wrap(~ variable): Creates multi-panel plots.

• geom_text(), paste0(): Adds annotations and labels to charts.

• scale_fill_manual(values = c(...)): Customizes color schemes for visualizations.

• theme_minimal(), axis.text.x = element_text(angle = 45, hjust = 1): Enhances chart appearance.

Table Display & Data Output

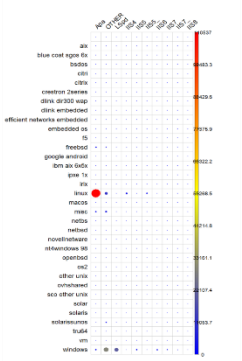
• kable(): Generates formatted summary tables.

• datatable(): Creates interactive tables with sorting and search functions.

3.3 Objective 3: To examine the correlation between web server types, operating systems, and ransom attack severity by analyzing attack patterns, downtime, and financial losses.

3.3.1 Analysis 1: Examining the relationship between OS and Web Server Vulnerability

Goal: To examine the relationship between operating systems and web servers to identify which combinations are most frequently targeted by ransom attacks.



The analysis of OS and Web Server vulnerability in ransomware attacks reveals clear patterns. The bar chart shows that Apache (Linux) experiences the highest number of attacks, while Windows-based IIS servers also face significant threats, though slightly lower. The heatmap further confirms that Linux-Apache combinations are highly targeted, with intense attack clusters and severe ransom demands. Meanwhile, other Unix-based systems (Solaris, FreeBSD) show fewer attacks, likely due to lower adoption rates. These findings suggest that attack patterns are strongly correlated with server popularity rather than inherent security flaws, emphasizing the need for enhanced security measures on widely used configurations.

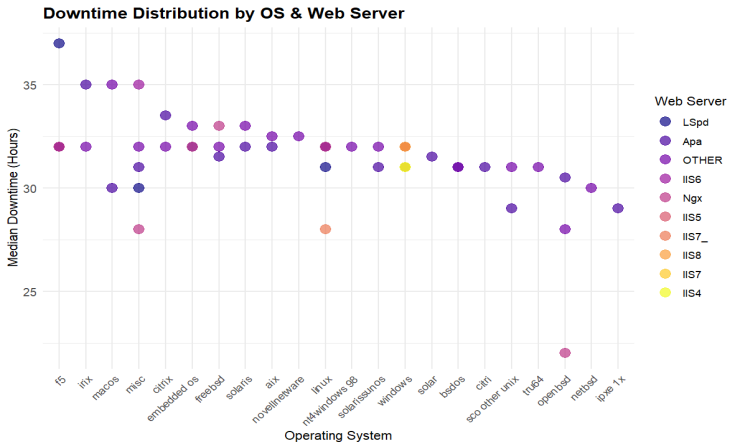
We have also ran Chi-squared test to further clarify our findings and the below is what we got:

```
        Pearson's Chi-squared test with simulated p-value (based on 2000 replicates)

data:  collapsed_os_web_table
X-squared = 162511, df = NA, p-value = 0.0004998
```

The Chi-Square test results confirm a strong correlation between Operating System (OS) and Web Server type in ransomware attacks (p-value < 0.001). This means that certain OS-Web Server combinations are disproportionately targeted, likely due to known vulnerabilities or widespread usage. The high Chi-Square value (162,511) suggests that attack patterns are not random, reinforcing that attackers focus on specific configurations. Organizations using highly targeted setups (e.g., Linux-Apache, Windows-IIS) should prioritize security measures like patching, monitoring, and system hardening to mitigate risks.

3.3.2 Analysis 2: Examining the relationship between OS, Web Server & Downtime

Goal: Identify if certain OS-Webserver combinations suffer longer downtimes.

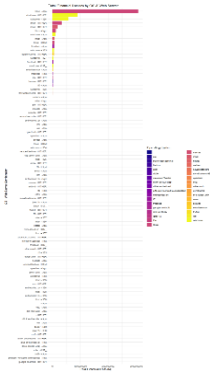


The scatter plot visualizes the distribution of median downtime across various operating systems and web servers. The data points show downtime clustering mostly between 30 and 35 hours, with no clear trend indicating that a particular OS-Web Server combination consistently experiences longer or shorter downtimes. While some variations exist, the spread appears relatively uniform across different web servers, suggesting no immediate correlation.

To validate this observation, a Kruskal-Wallis test was performed after assessing normality using the Shapiro-Wilk test, which indicated that the downtime data was not normally distributed. The test produced a chi-squared value of 78.764 with 90 degrees of freedom and a p-value of 0.7953. Since the p-value is high, there is no statistically significant difference in downtime across OS-Web Server combinations.

3.3.3 Analysis 3: Examining the relationship between OS, Web Server & Financial Losses

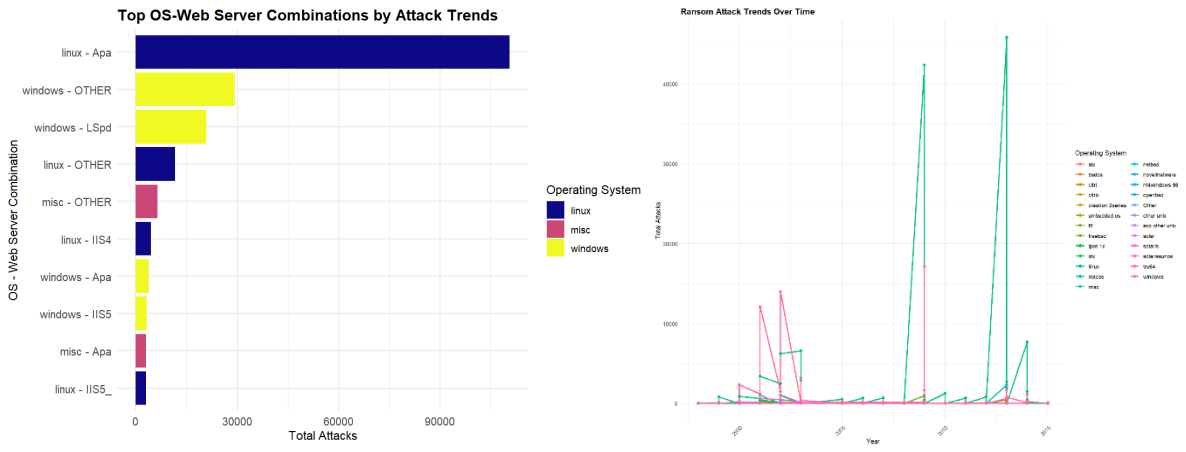
Goal: Examine which OS-WebServer combinations lead to the highest financial losses due to ransom attacks.



The bar chart illustrates the total financial losses associated with various OS-Web Server combinations due to ransomware attacks. The most significant observation is that Linux-Apache incurs the highest financial losses, surpassing all other combinations by a wide margin. This suggests that Linux systems running Apache web servers are prime targets for ransomware attacks, potentially due to their prevalence in enterprise and web hosting environments. Following closely, Windows-based systems also show substantial financial losses, particularly combinations involving IIS and other web servers, indicating that Windows environments remain highly vulnerable to costly attacks.

Beyond these dominant categories, other OS-Web Server combinations, including miscellaneous configurations and FreeBSD-based systems, contribute to financial losses, though at much lower levels. The data suggests that Linux and Windows environments are disproportionately affected, possibly due to their widespread adoption in high-value infrastructures. Interestingly, web servers categorized as “OTHER” also account for significant losses, implying that non-mainstream or custom configurations are not necessarily safer.

3.3.4 Analysis 4: Examining the relationship between OS, Web Server & Attack Trends Over Time

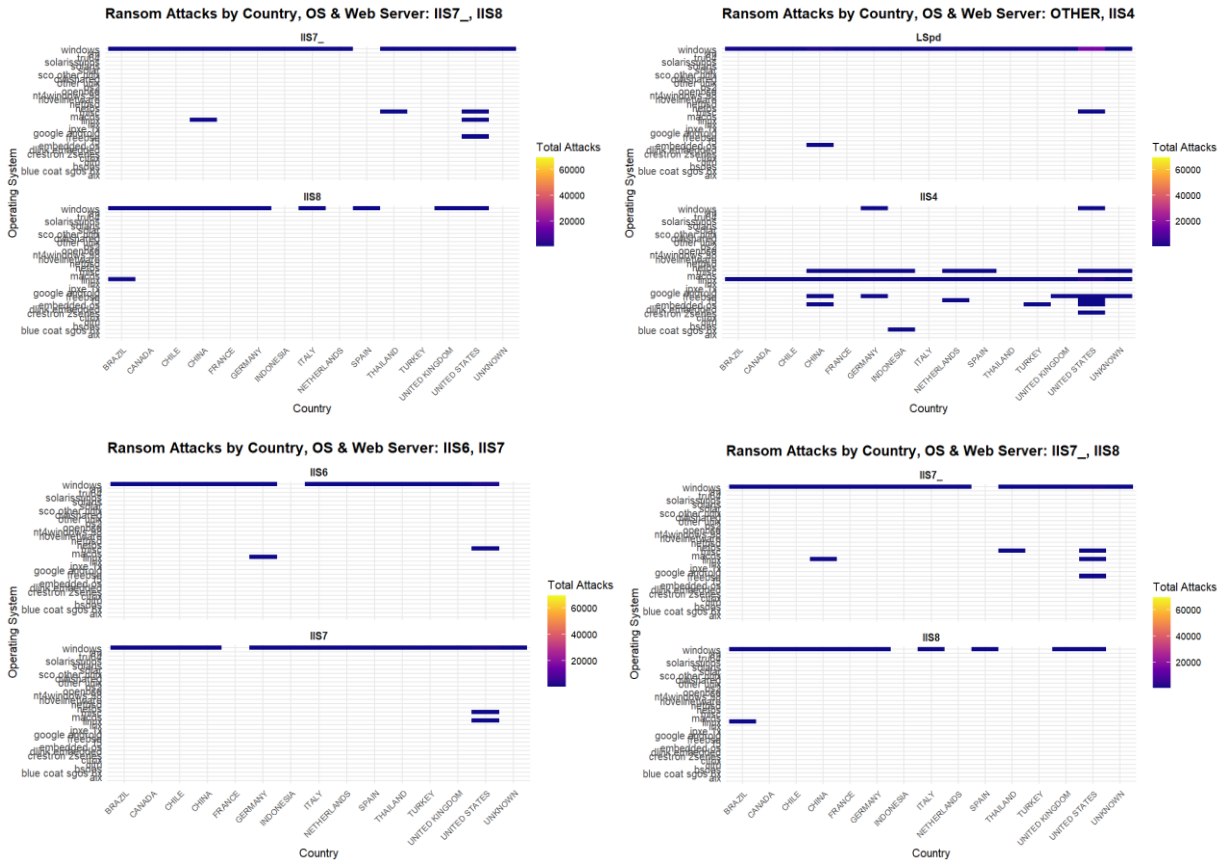


The analysis of OS-Web Server attack trends reveals that Linux with Apache is the most frequently targeted combination, significantly surpassing other configurations in attack volume. Windows-based systems, particularly those labeled as "OTHER" and those using LiteSpeed or IIS, also experience substantial attacks but at a lower magnitude. The second visualization,

depicting ransom attack trends over time, indicates periodic spikes in attack activity, particularly in the late 1990s and early 2010s. These spikes suggest moments of heightened vulnerability or large-scale coordinated attacks. The data highlights the persistent risk to Linux-based web servers, reinforcing the need for enhanced security measures, particularly for Apache configurations. Meanwhile, Windows and miscellaneous systems also remain attractive targets, emphasizing the importance of consistent monitoring and proactive security updates to mitigate vulnerabilities.

3.3.5 Analysis 4: Examining the relationship between OS, Web Server, Ransom Attacks & Country-Specific Trends

The above is an overview of attacks based off countries; we will take a closer look below with webserver specific attacks on countries.



The previous four visualizations highlighted ransom attack distributions across different OS and web server combinations. IIS-based web servers (IIS5, IIS6, IIS7, and IIS8) were frequent targets, with Windows OS being the most affected across various countries, particularly in the

United States, Turkey, and the United Kingdom. Attack patterns showed some clustering, indicating that certain regions faced more sustained threats, especially on outdated or widely deployed IIS versions. Meanwhile, miscellaneous and lesser-known OS-web server combinations had scattered attacks, suggesting opportunistic exploitation rather than targeted campaigns.

The latest visualization, focusing on Apache (Apa), LiteSpeed (LSpd), and OTHER web servers, demonstrates a broader distribution of ransom attacks across multiple operating systems and countries. Unlike IIS-based servers, which showed regional concentration, attacks on Apache and LiteSpeed appear more widespread, affecting Linux, macOS, and even embedded systems. The presence of high attack frequencies in Linux-based web servers suggests that Apache remains a prime target due to its extensive use. Additionally, the OTHER category shows a diverse attack surface, covering Google Android, Solaris, BSD variants, and embedded OS, reinforcing the fact that attackers are actively scanning and compromising multiple system types beyond traditional Windows-based IIS deployments.

3.3.6 Conclusion

The findings from this analysis highlight clear patterns in ransomware attack trends, emphasizing the significant role of OS and web server configurations in determining vulnerability levels. The Linux-Apache combination emerges as the most frequently targeted setup, experiencing the highest number of attacks, financial losses, and ransom demands. Windows-based IIS servers also remain highly vulnerable, particularly in regions such as the United States, Turkey, and the United Kingdom, where attacks cluster around outdated IIS versions. While downtime does not

show a statistically significant variation across OS-Web Server combinations, the financial impact and attack frequency reveal that widely adopted configurations attract more cyber threats.

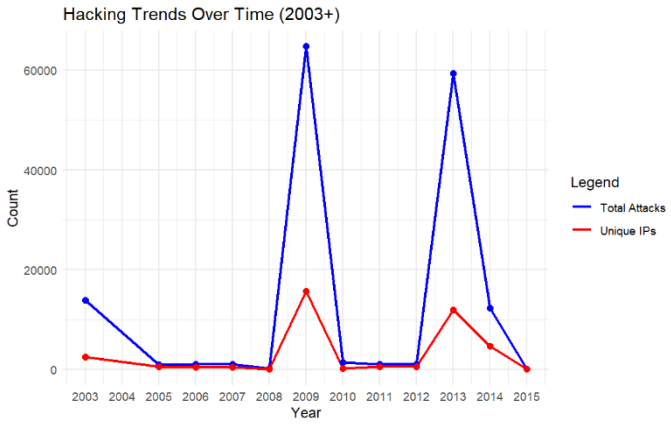
Furthermore, the examination of country-specific attack trends confirms that both mainstream and alternative web server configurations are at risk, with Apache and LiteSpeed showing a more globally dispersed attack pattern, while Windows-IIS remains regionally concentrated. This underscores the importance of proactive cybersecurity measures such as regular patching, monitoring, and system hardening to mitigate risks. Organizations running highly targeted setups must adopt multi-layered security approaches, including advanced threat detection, secure backups, and incident response planning, to reduce their exposure to ransomware attacks.

3.4 Objective 4: To analyze the trends of hacking activities in recent years and identify the most prevalent attack types, along with their associated vulnerabilities and impacts. (Ong Kai Ying TP086065)

3.4.1 What causes fluctuations in hacking incidents since 2003, and how do unique IP trends reflect changes in attacker behavior?

```
> yearly_stats <- df %>%group_by(Year) %>%summarise(
+     Total_Attacks = n(),Unique_IPs = n_distinct(IP))
>
> print(yearly_stats)
# A tibble: 12 x 3
    Year Total_Attacks Unique_IPs
   <int>         <int>      <int>
 1  2003         13865       2458
 2  2005          1000        556
 3  2006          1008        424
 4  2007          1030        432
 5  2008           171        130
 6  2009         64786      15701
 7  2010          1433        256
 8  2011          1013        605
 9  2012          1061        559
10  2013         59351      11964
11  2014         12358       4648
12  2015           136        105
```

```
df$Date <- as.Date(df$Date, format = "%Y-%m-%d")
df <- df[!is.na(df$Date), ]
df$Ye
ar <- as.integer(format(df$Date, "%Y"))
df <- df %>% filter(Year >= 2003)

incident_count <- df %>%
  group_by(Year) %>%
  summarise(Incidents = n())

unique_ips <- df %>%
  group_by(Year) %>%
  summarise(Unique_IPs = n_distinct(IP))

attack_trends <- merge(incident_count, unique_ips, by="Year")

library(ggplot2)
ggplot(attack_trends, aes(x = Year)) +
  geom_line(aes(y = Incidents, color = "Total Attacks"), size = 1) +
  geom_line(aes(y = Unique_IPs, color = "Unique IPs"), size = 1) +
  geom_point(aes(y = Incidents), color = "blue", size = 2) +
  geom_point(aes(y = Unique_IPs), color = "red", size = 2) +
  scale_color_manual(values = c("Total Attacks" = "blue", "Unique IPs" = "red")) +
  scale_x_continuous(breaks = seq(2003, max(df$Year), by = 1)) +
  labs(title = "Hacking Trends Over Time (2003+)",
       x = "Year",
       y = "Count",
       color = "Legend") +
  theme_minimal()
```



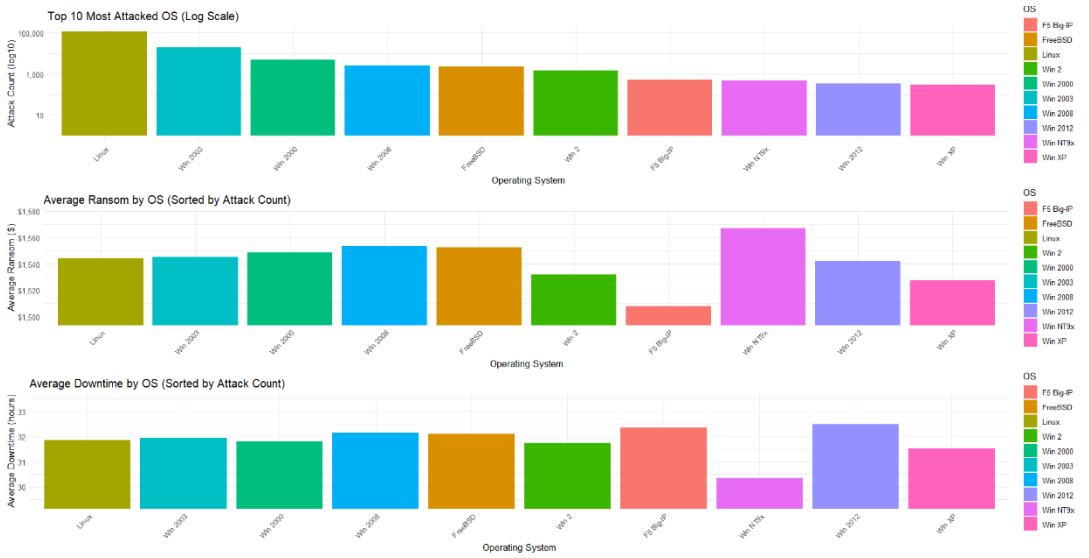
From the graph, we observe that fluctuations in attack incidents (blue line) and the number of unique IPs (red line) do not show a consistent upward or downward trend but rather sharp spikes in certain years followed by declines.

In 2003, hacking activity was high with 13,865 attacks and 2,458 unique IPs. From 2005 to 2008, attacks declined, hitting a low of 171 in 2008. In 2009, incidents surged to 64,786 attacks and 15,701 unique IPs. From 2010 to 2012, attacks remained low, though unique IPs peaked in 2011. Another spike occurred in 2013 (59,351 attacks, 11,964 unique IPs), but in 2014-2015, both metrics dropped sharply, reaching a record low of 136 attacks and 105 unique IPs in 2015.

As conclusion, hacking incidents spiked in 2003, 2009, and 2013, likely due to new attack tools and global events. Drops in 2005-2008 and 2014-2015 suggest better cybersecurity or changing tactics. Unique IP trends show shifts in attack scale and botnet use, stressing the need for ongoing security improvements.

3.4.2 Which operating systems face the highest attack frequency, and which OS has the most ransomware incidents and the highest financial loss?

```
p1 <- ggplot(attack_count, aes(x = reorder(OS, -Attack_Count), y = Attack_Count, fill = OS)) +
  geom_bar(stat = "identity") +
  scale_y_log10(labels = comma) +
  labs(title = "Top 10 Most Attacked OS (Log Scale)", x = "Operating System", y = "Attack Count (log10)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p2 <- ggplot(avg_ransom, aes(x = reorder(OS, -Attack_Count), y = Average_Ransom, fill = OS)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(labels = dollar, limits = c(min(avg_ransom$Average_Ransom) - 10, max(avg_ransom$Average_Ransom) + 10), oob = scales::rescale_none) +
  labs(title = "Average Ransom by OS (Sorted by Attack Count)", x = "Operating System", y = "Average Ransom ($)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

p3 <- ggplot(avg_downtime, aes(x = reorder(OS, -Attack_Count), y = Average_Downtime, fill = OS)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(labels = comma, limits = c(min(avg_downtime$Average_Downtime) - 1, max(avg_downtime$Average_Downtime) + 1), oob = scales::rescale_none) +
  labs(title = "Average Downtime by OS (Sorted by Attack Count)", x = "Operating System", y = "Average Downtime (hours)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grid.arrange(p1, p2, p3, ncol = 1)
```



The bar chart shows Linux as the most attacked OS, with more than 100000 attacks, far more than Windows Server 2003 more than attacks. Other Windows systems, like Windows 2000 and Windows 2008, had moderate attacks. Unix-based systems, such as FreeBSD, F5 Big-IP had fewer attacks.

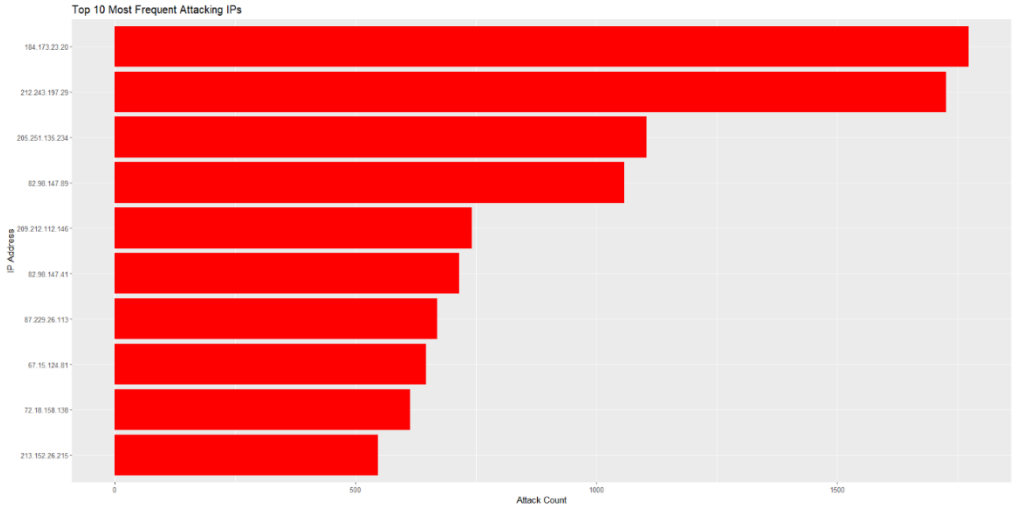
Linux is the top target in this dataset. Older Windows systems, like Windows 2000,2008, especially the Window N T9x had low number of attacks but average Ransome are the higher. Window system make up a large portion of attacks, showing they are key ransomware targets.

For downtime and ransom, Window 2012 had low number of attacks but the longest downtime which is around 32 hours. Windows N T9x had the highest ransom demands, but downtime are the lowest, around 30.5 hour.

As conclusion, Linux is the most targeted OS, old Windows systems are prime ransomware targets, suggest that outdated systems with known vulnerabilities are prime targets. Longer downtimes in niche OS, systems like FreeBSD and F5 Big-IP might have limited support or recovery options, leading to longer recovery times.

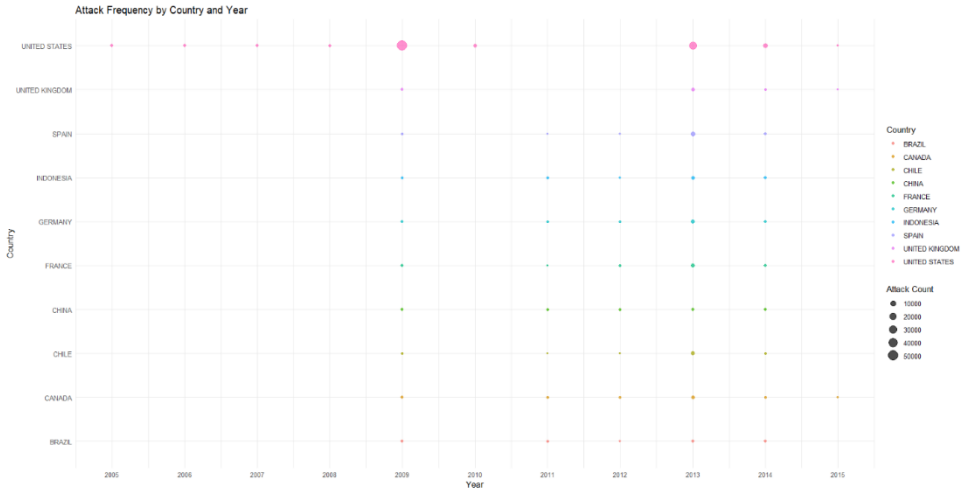
3.4.3 Are there specific IPs that repeatedly attack within a short period, indicating potential DDoS or brute-force attempts?

```
data$Date <- as.Date(data$Date, format = "%Y-%m-%d")
data <- data %>% filter(Date >= as.Date("2005-01-01"))

#1. Plot: Top 10 Most Frequent Attacking IPs
top_ips <- data %>%
  filter(IP != "" & !is.na(IP)) %>%
  count(IP, sort = TRUE) %>%
  top_n(10, n)

ggplot(top_ips, aes(x = reorder(IP, n), y = n)) +
  geom_bar(stat = "identity", fill = "red") +
  coord_flip() +
  labs(title = "Top 10 Most Frequent Attacking IPs", x = "IP Address", y = "Attack Count")
```



To identify DDoS attack trends, the first step is to identify the IPs with the most aggressive attackers. This bar chart shows that the IP 184.173.23.20 has the highest attack frequency, with over 1,750 attacks. The second most frequent attacker is 212.243.197.29, with more than 1,500 attacks.

```
data <- data %>%
  mutate(Year = year(Date)) %>%
  filter(Year >= 2005)

top_countries <- data %>%
  count(Country, sort = TRUE) %>%
  top_n(10, n) %>%
  pull(Country)

filtered_data <- data %>%
  filter(Country %in% top_countries) %>%
  count(Year, Country)

heatmap_data <- acast(filtered_data, Country ~ Year, value.var = "n", fill = 0)

ggplot(filtered_data, aes(x = Year, y = Country, size = n, color = Country)) +
  geom_point(alpha = 0.7) +
  scale_x_continuous(breaks = seq(2005, max(filtered_data$Year), 1)) +
  labs(title = "Attack Frequency by Country and Year", x = "Year", y = "Country", size = "Attack Count") +
  theme_minimal()
```

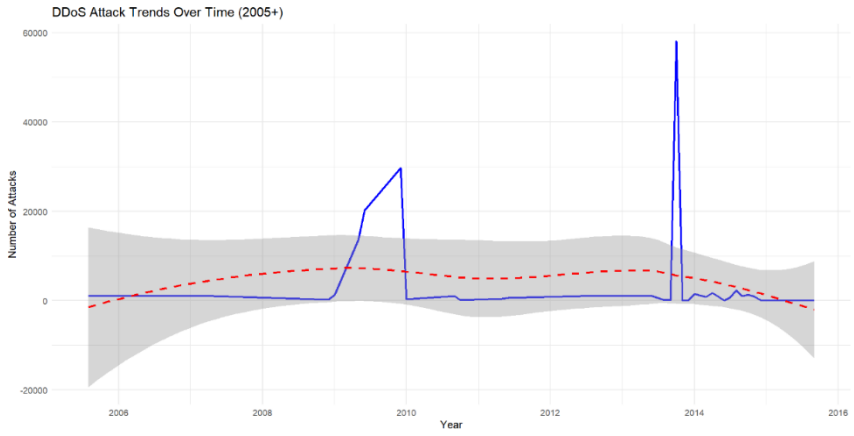


The second step is to identify the attack waves and affected countries. In the point chart, we can see that the US is the country experiencing the most attacks. In 2009, the attack count was higher, reaching around 500,000.

```
clean_data <- data %>%
  filter(!is.na(IP) & IP != "Unknown",
         !is.na(Country) & Country != "Unknown",
         !is.na(Downtime),
         as.integer(format(as.Date(Date, "%Y-%m-%d"), "%Y")) >= 2005)

top_countries <- clean_data %>%
  count(Country, sort = TRUE) %>%
  slice_head(n = 10) %>%
  pull(Country)

filtered_data <- clean_data %>%
  filter(Country %in% top_countries)

top_ips <- filtered_data %>%
  group_by(IP, Country) %>%
  summarise(attack_count = n(), total_downtime = sum(Downtime, na.rm = TRUE)) %>%
  arrange(desc(attack_count)) %>%
  slice_head(n = 10)

ggplot(top_ips, aes(x = attack_count, y = total_downtime, color = Country, label = IP)) +
  geom_point(size = 4) +
  geom_text(vjust = -0.5, hjust = 0.5, size = 3) +
  scale_x_continuous(limits = c(min(top_ips$attack_count) * 0.9, max(top_ips$attack_count) * 1.1)) +
  scale_y_continuous(limits = c(min(top_ips$total_downtime) * 0.9, max(top_ips$total_downtime) * 1.1)) +
  labs(title = "Top 10 High-Impact Attackers (2005+): Downtime vs. Attack Count",
       x = "Number of Attacks",
       y = "Total Downtime (minutes)",
       color = "Country") +
  theme_minimal()
```

The third step is to identify which countries the IPs belong to. The IP 184.173.23.20 has the highest attack count, and it is mostly found in the US. This is followed by 212.243.197.29 and 205.251.135.234, both of which are also located in the US. This indicates that a significant portion of the attacks originate from within the country.

```
data <- data %>% filter(Date >= as.Date("2005-01-01"))
monthly_attacks <- data %>%
  mutate(YearMonth = floor_date(Date, "month")) %>%
  count(YearMonth)

ggplot(monthly_attacks, aes(x = YearMonth, y = n)) +
  geom_line(color = "blue", size = 1) +
  geom_smooth(method = "loess", color = "red", linetype = "dashed") +
  labs(title = "DDoS Attack Trends Over Time (2005+)",
       x = "Year", y = "Number of Attacks") +
  theme_minimal()
```



Based on the analysis from the previous three steps, we can easily identify attack spikes and trends over time using a line graph. The blue line represents the number of DDoS attacks per year, while

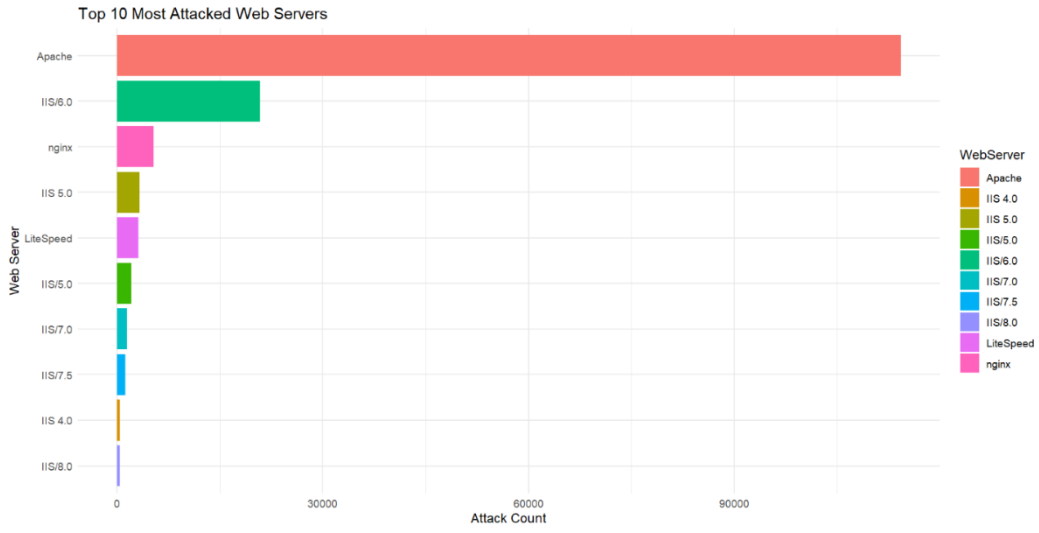
the red dashed line is a smoothed trend line that shows the overall pattern of attacks over time. The gray shaded area represents the confidence interval, indicating the uncertainty in trend prediction.

A noticeable spike occurs between 2009 and 2010, where the number of attacks increases rapidly before dropping back down. In 2014, another extreme peak is observed, with the number of attacks surging to over 60,000 before quickly returning to normal levels. The sudden surge in attacks during these periods suggests several possible contributing factors. One easy explanation is major global DDoS attack campaigns, which may have led to a sharp rise in attack frequency.

As a conclusion to this analysis, we observe that DDoS attacks and brute-force attacks are mostly concentrated in the U.S., with two specific IP addresses, 184.173.23.20 and 212.243.197.29. These attack waves often align with major global cyber events, showing the impact of geopolitical factors and evolving cybercriminal tactics.

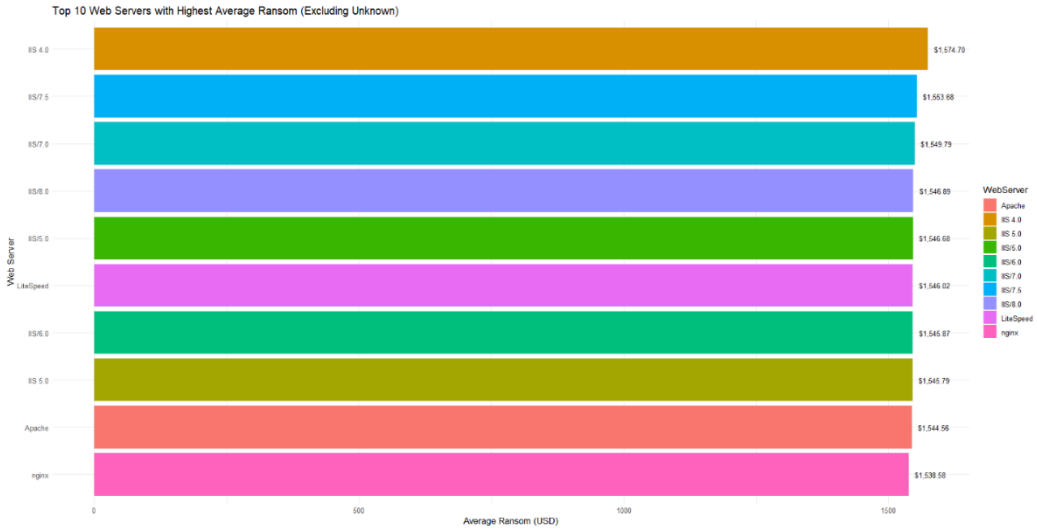
3.4.4 Which web servers are most frequently targeted in attacks, and what factors contribute to the varying ransom demands and financial losses associated with these attacks?

```
ggplot(attack_counts, aes(x = reorder(WebServer, Attack_Count), y = Attack_Count, fill = WebServer)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Most Attacked Web Servers (Excluding Unknown)", x = "Web Server", y = "Attack Count") +
  theme_minimal()
```



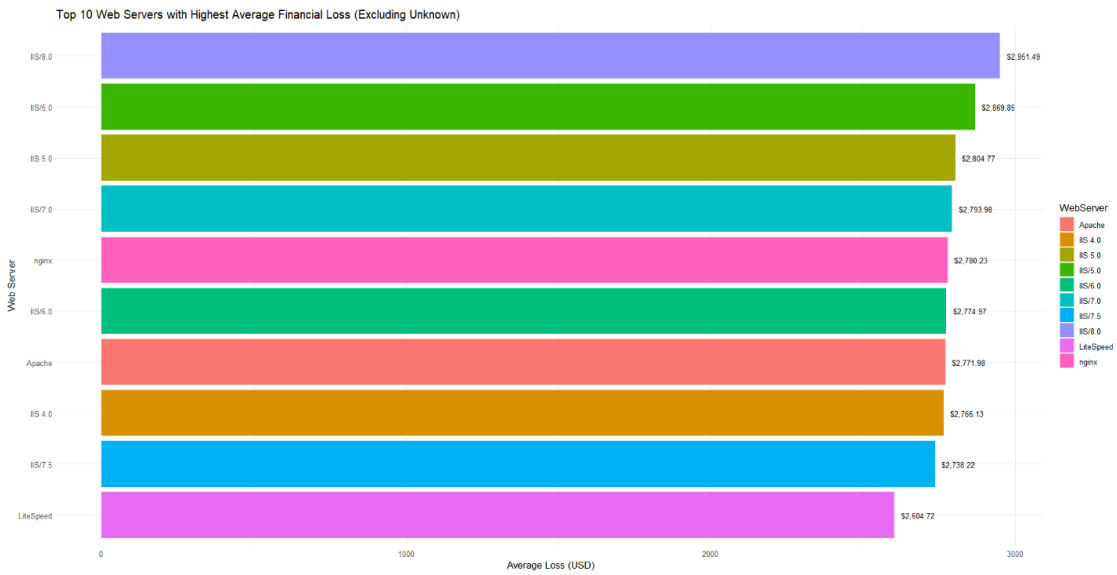
By referring to the barchart, we observed that Apache is the most attacked web server, which is holds 114243 of attacks,then following by the IIS/6.0 which is 20851 attacks.

```
ggplot(ransom_summary, aes(x = reorder(WebServer, Average_Ransom), y = Average_Ransom, fill = WebServer)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Web Servers with Highest Average Ransom (Excluding Unknown)",
       x = "Web Server", y = "Average Ransom (USD)") +
  theme_minimal() +
  geom_text(aes(label = scales::dollar(Average_Ransom)), hjust = -0.2, size = 3)
```



The average ransom is fairly consistent across web server, which is around $1545-1575. IIS4.0 show the highest average ransom at $1574.70, while ngix is slightly lower at $1538.58.

```
ggplot(loss_summary, aes(x = reorder(WebServer, Average_Loss), y = Average_Loss, fill = WebServer)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Web Servers with Highest Average Financial Loss (Excluding Unknown)",
       x = "Web Server", y = "Average Loss (USD)") +
  theme_minimal() +
  geom_text(aes(label = scales::dollar(Average_Loss)), hjust = -0.2, size = 3)
```



IIS/8.0 has the highest average lost which is $2951.49, the followed by the IIS/5.0 at $2870, while LiteSpeed has the lowest average loss of $2604.73.

After observe this 3 of bar chart, we conclude that even though Apache has highest attack count but does not have highest ransom or financial loss. This might indicate that it is an easy target, but attackers do not necessarily extract the highest ransoms or cause the largest financial damage per attack. IIS/8.0 holds a lower attack count, but has highest financial loss, which could suggest that while it may not be an easy target for frequent attacks, the attacks are more costly or damaging when they occur.

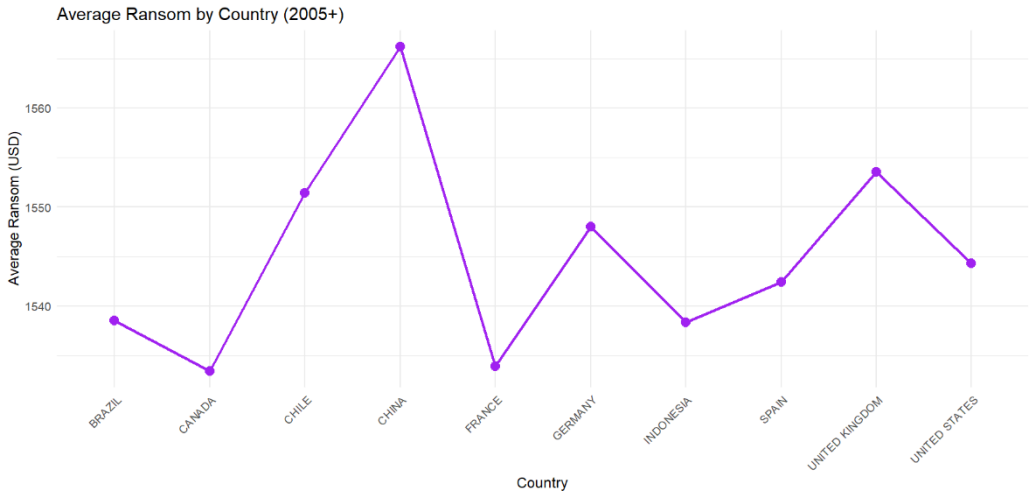
As conclusion, IIS versions generally have a higher Average Loss than Apache and Nginx. This might be due to different weaknesses in these web servers or how attackers target them.

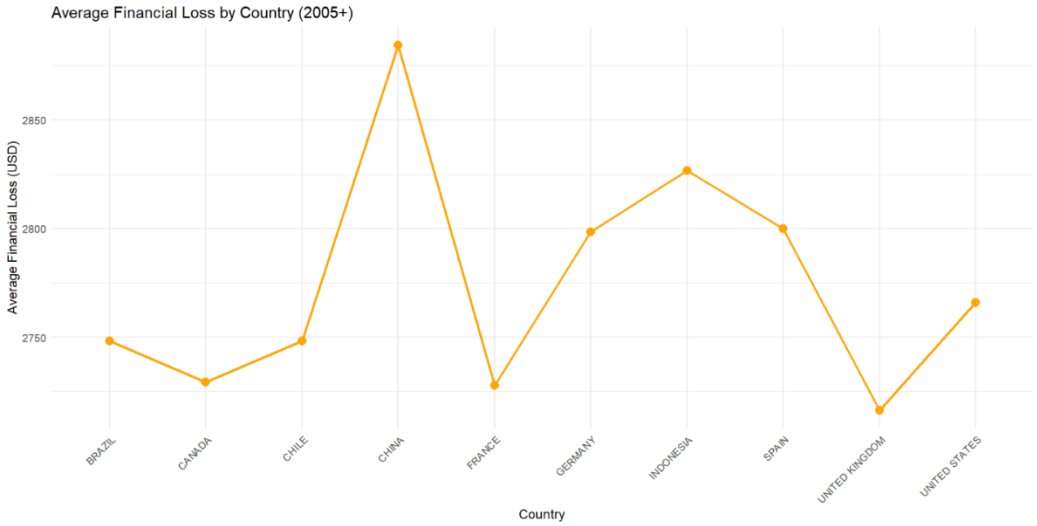
Additional Feature

3.4.5 Which country experiences the largest financial losses due to ransom demands?

	Country	attack_count	unique_ips	total_downtime	avg_ransom	avg_loss
	<chr>	<int>	<int>	<int>	<dbl>	<dbl>
1	UNITED STATES	89817	21205	2861371	1544.	2766.
2	SPAIN	5931	282	190202	1542.	2800.
3	GERMANY	4653	1354	147933	1548.	2798.
4	FRANCE	4026	490	129628	1534.	2728.
5	CHILE	2501	169	80371	1551.	2748.
6	UNITED KINGDOM	2394	728	76499	1554.	2716.
7	CANADA	2326	524	74978	1533.	2729.
8	INDONESIA	2268	483	72396	1538.	2827.
9	BRAZIL	2015	669	63941	1539.	2748.
10	CHINA	1886	962	58234	1566.	2884.

```
ggplot(country_impact, aes(x = Country, y = avg_ransom, group = 1)) +
  geom_line(color = "purple", size = 1) +
  geom_point(color = "purple", size = 3) +
  labs(title = "Average Ransom by Country (2005+)",
       x = "Country",
       y = "Average Ransom (USD)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggplot(country_impact, aes(x = Country, y = avg_loss, group = 1)) +
  geom_line(color = "orange", size = 1) +
  geom_point(color = "orange", size = 3) +
  labs(title = "Average Financial Loss by Country (2005+)",
       x = "Country",
       y = "Average Financial Loss (USD)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```





The U.S. is the most attacked country, with 89,817 attacks. More attacks usually mean more downtime, and the U.S. faces the longest disruptions. Financial losses (ransom and damage) are similar across countries, with small differences. The U.S. has far more attacks than others. Spain loses more money per attack, meaning the attacks are stronger. China has fewer attacks but high losses, showing they target big companies. Countries with many unique IPs (Germany, UK, China) likely face botnets or global attacks. Spain and Indonesia have fewer attacks but higher losses, meaning they hit harder. Brazil and Chile take longer to recover, leading to more downtime.

In conclusion, the U.S. gets the most attacks and suffers the most. Spain and Indonesia lose more money due to stronger attacks. China faces fewer but costly attacks. Germany, the UK, and China see many unique IPs, likely from botnets. Among the top 10 attacked countries, Spain faces high financial losses from ransomware, while China sees fewer attacks but greater losses, indicating targeted strikes. Germany, the UK, and Canada have high unique IP counts, suggesting botnet

activity. Indonesia, Brazil, and Chile experience prolonged downtimes, likely due to weaker cybersecurity.

Reason to be additional feature:

This feature shows how attacks, losses, and downtime vary by country. It highlights the U.S. as the most attacked, Spain and China as high-loss countries, and explains how unique IPs suggest global attacks. It also analyzes the top 10 most attacked countries and the types of attacks they face, such as ransomware, data breaches, and DDoS attacks, providing deeper insights into global hacking patterns.

Overall, hacking trends show clear patterns, with major spikes related to new attack tools, botnet activity, and geopolitical events. Strengthening cybersecurity, especially for legacy systems and high-risk industries, is key to reducing future threats.

Conclusion

Overall Discussion:

In addition to the observed disparities in ransom attack frequency and downtime across countries and web servers, the analysis highlights how the resilience of web servers and the strength of national cybersecurity frameworks significantly influence recovery times. Countries with robust cybersecurity protocols and well-maintained infrastructure experience shorter downtimes, while those with weaker defenses are more vulnerable to prolonged disruptions. The study also reinforces that outdated operating systems not only contribute to higher losses but also serve as prime targets for cybercriminals exploiting known vulnerabilities. Rapid adaptation to evolving ransomware tactics is crucial to maintaining strong defenses.

Additional Recommendation:

Apart from updating web servers and operating systems, organizations should conduct regular cybersecurity audits to identify potential risks and patch vulnerabilities proactively. Investing in advanced threat detection systems powered by artificial intelligence and machine learning can help predict and mitigate potential threats in real time. Collaboration with global cybersecurity communities can also enhance threat intelligence sharing, contributing to faster identification of emerging ransomware trends.

Expanded Limitation and Future Direction:

A critical limitation of this analysis is the potential underestimation of ransomware impacts due to unreported or undetected attacks. Many organizations, especially in regions with weaker regulatory requirements, might not disclose ransomware incidents, leading to gaps in the data. Future research could explore the role of automation in cyberattacks, such as AI-driven malware, and assess how new technologies like quantum computing might alter the threat landscape. Developing predictive models to assess which regions or sectors might face increased risks could also be a valuable direction for future work.

Workload Matrix

Name	Tasks

Manreen Kaur A/P Jagjit Singh	• Hypothesis
	• Objective 1
	• Data Preparation
	  ➢ Data import
	  ➢ Cleaning /pre-processing
	  ➢ Data Validation
	• Data Analysis 1-6 (3.1.1-3.1.6)
	• Conclusion for analysis
	• Conclusion
	• Additional Features
	• References

Lim Eazen	• Introduction • Assumptions • Techniques • Objective 2 • Data Preparation ➢ Cleaning /pre-processing • Data Analysis 1-5 (3.2.1-3.2.5) • Conclusion for analysis • Key insights • Conclusion • Additional Features
Brayden Yoong Policarpio	• Objective 3 • Data Preparation ➢ Cleaning /pre-processing • Data Analysis 1-5 (3.3.1-3.3.5) • Conclusion for analysis • Additional Features

Ong Kai Ying	• Objective 4
	• Data Preparation
	  ➢ Data import
	  ➢ Cleaning /pre-processing
	  ➢ Data Validation
	• Data Analysis 1-5 (3.4.1-3.4.5)
	• Conclusion for analysis
	• Additional Features
	• References

Reference

1. GeeksforGeeks. (2023, July 5). Data cleaning in R. GeeksforGeeks. https://www.geeksforgeeks.org/data-cleaning-in-r/
2. Holtz, Y. (n.d.). The R Graph Gallery – Help and inspiration for R charts. The R Graph Gallery. https://r-graph-gallery.com/
3. Bobbitt, Z. (2024b, April 26). How to Use geom_text() in R. Statology. https://www.statology.org/r-geom_text/
4. Coder, R. (2021, April 9). Combining plots in R. R CHARTS | a Collection of Charts and Graphs Made With the R Programming Language. https://r-charts.com/base-r/combining-plots/
5. A.Long, J. (2024). Plot simple effects in regression models - effect_plot. Retrieved from jtools: https://jtools.jacob-long.com/reference/effect_plot
6. Wickham, H. (2023). Simple Data Frames. Retrieved from Tidyverse.org https://tibble.tidyverse.org
7. Wickham, H. (2023). Simple Data Frames. Retrieved from Tidyverse.org https://r-graph-gallery.com/lollipop-plot.html
8. Holtz, Y. (n.d.). Grouped and Stacked barplot | the R Graph Gallery. Retrieved from r-graph-gallery.com https://r-graph-gallery.com/stacked-barplot.html
9. Plot simple effects in regression models — effect_plot. (2024). Jacob-Long.com. https://jtools.jacob-long.com/reference/effect_plot
10. Correlation coefficient and correlation test in R. (n.d.). Stats and R. https://statsandr.com/blog/correlation-coefficient-and-correlation-test-in-r/#correlation-coefficient
11. Rays, S. (2024, November). Introduction to Statistical Testing in R Part 5— Correlation & Regression. Medium. https://medium.com/@serurays/introduction-to-statistical-testing-in-r-part-5-correlation-regression-34bb8a2f91aa