Smt . Durgadevi Sharma Charitable Trust's

Chandrabhan Sharma College

of Arts, Commerce & Science

(Autonomous)

Hindi Linguistic Minority Institution

(Affiliated to University of Mumbai)

NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)

Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

Subject Name : **Soft Computing Techniques & Data Science & Security Breaches and Counter Measures**

Name : **PRATIK JAGDISHCHANDRA MAURYA**

Seat No : **1310226**

Teaching Faculty : **Mr. Arvind Singh & Mr. Sandeep Vishwakarma & Nitesh Shukla**

DEPARTMENT OF INFORMATION TECHNOLOGY

CHANDRABHAN SHARMA COLLEGE OF ARTS,

COMMERCE & SCIENCE

NAAC RE-ACCREDITED 'A' Grade (CGPA 3.10)

(Autonomous)

MUMBAI, 400076

MAHARASHTRA

2024-2025

Smt . Durgadevi Sharma Charitable Trust's
Chandrabhan Sharma College
of Arts, Commerce & Science
(Autonomous)
Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

Subject Name : **Soft Computing Techniques**

Name : **PRATIK JAGDISHCHANDRA MAURYA**

Seat No : **1310226**

Teaching Faculty : **Mr. Arvind Singh**

DEPARTMENT OF INFORMATION TECHNOLOGY

CHANDRABHAN SHARMA COLLEGE OF ARTS,

COMMERCE & SCIENCE

NAAC RE-ACCREDITED 'A' Grade (CGPA 3.10)

(Autonomous)

MUMBAI, 400076

MAHARASHTRA

2024-2025

Smt . Durgadevi Sharma Charitable Trust's
Chandrabhan Sharma College
of Arts, Commerce & Science
(Autonomous)
Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that Mr./~~Miss~~ **PRATIK JAGDISHCHANDRA MAURYA** having Exam SeatNo. **1310226** of M.Sc.IT **(Semester I)** has complete the Practical work in the subject of "**Soft Computing Techniques**" during the **Academic year 2024-25** under the guidance of **Mr. Arvind Singh** being the Partial requirement for The fulfilment of the curriculum of Degree of Master of Science in Information Technology, University of Mumbai.

Internal Examiner **Co-Ordinator**

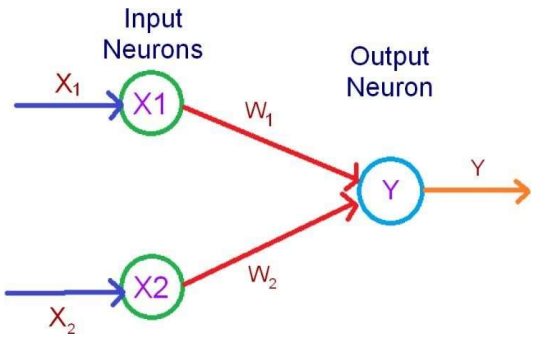
Date: **College Seal**

External Examiner

INDEX

b	Write a program for Hopfield network model for associative memory		
8.	**Implement the Following**		
a	Membership and Identity Operators | in, not in,		
b.	Membership and Identity Operators is, is not		
9.	**Implement the Following**		
a	Find ratios using fuzzy logic		
b	Solve Tipping problem using fuzzy logic		
10.	**Implement the Following**		
a	Implementation of Simple genetic algorithm		
b	Create two classes: City and Fitness using Genetic algorithm		

Practical 1-A
Design a simple linear neural network model



The above diagram represents a simple neural network, having one input and one output layer; there are two neurons at the input layer and one at the output.
The input and the output layers are interconnected by weights, we are not considering the bias (b = 0)
The inputs are x_1 and x_2 to the neurons X1 and X2 respectively
The neuron X1 is connected by weight w_1 and X2 by weight w_2 to the output neuron respectively.
The threshold value for activation is 0 as represented by the activation function below

The net input to the output is

$$y_{in} = x_1w_1 + x_2w_2$$

In the present case we use the following activation for the output neuron

$$Y_{out} = \begin{cases} 0 & \text{if } y_{in} <= 0 \\ 1 & \text{if } y_{in} > 0 \end{cases}$$

The following Python program shows the implementation of the simple neural network

```
print("Enter Value of X1 =")
x1 = int(input())

print("Enter Value of X2 =")
x2 = int(input())

print("Enter Value of W1 =")
w1 = int(input())

print("Enter Value of W2 =")
w2 = int(input())
```

```
yin = x1*w1 + x2*w2
print("Net Input to the Output Neuron=",yin)

if yin <= 0:
      yout = 0
else :
      yout = 1

print("The Output of given neural network = ",yout)
```

The output is

```
Enter Value of X1 =
-1
Enter Value of X2 =
11
Enter Value of W1 =
1
Enter Value of W2 =
1
Net Input to the Output Neuron= 10
The Output of given neural network = 1
```

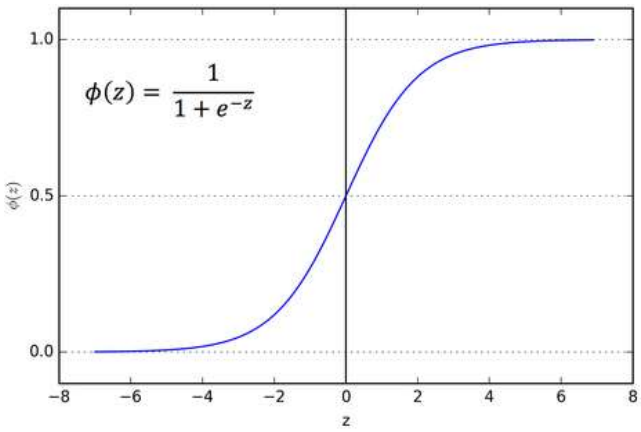
Practical 1-B

Neural net using both binary and bipolar sigmoidal function.

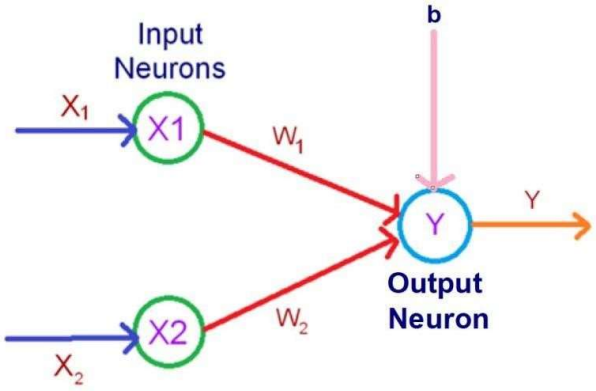
Part 1: Using Binary sigmoidal function

The function takes any real value as input and outputs values in the range 0 to 1. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to 0.0.

The Sigmoid Function curve looks like a S-shape.



Consider the following Neural Network



The Python code for the given case is as follows

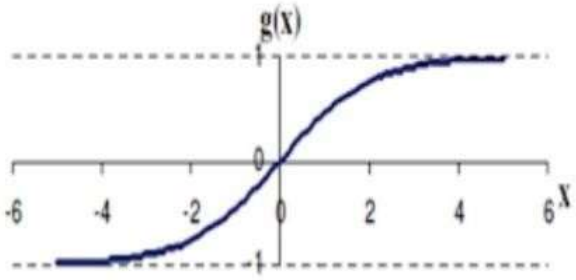
```
import numpy as np

print("Enter Value of X1 =");
x1 = int(input());

print("Enter Value of X2 =");
x2 = int(input());

print("Enter Value of W1 =");
w1 = int(input());

print("Enter Value of W2 =");
w2 = int(input());

b = 1;

yin = b + (x1*w1 + x2*w2);
print("Net Input to the Output Neuron=",yin);

output = 1/(1 + np.exp(-yin));
print("Output=", round(output, 4));
```

The output is as follows

```
Enter Value of X1 =
 -1
Enter Value of X2 =
2
Enter Value of W1 =
-2
Enter Value of W2 =
-1
Net Input to the Output Neuron= 1
Output= 0.7311
```

Irrespective of the input, the output is always between 0 and 1

Part 2: Using Bipolar sigmoidal function
The Bipolar sigmoidal function is represented in the following way



Mathematically it is given by

$$y = f(y_{in}) = \frac{2}{1 + e^{-y_{in}}} - 1$$

The python code for bipolar sigmoidal function is given by

```
import numpy as np
print("Enter Value of X1 =")
x1 = int(input())
print("Enter Value of X2 =")
x2 = int(input())
print("Enter Value of W1 =")
w1 = int(input())
print("Enter Value of W2 =")
w2 = int(input())
b = 1
yin = b + (x1*w1 + x2*w2)
print("Net Input to the Output Neuron=", yin)
```

The output is as follows

```
Enter Value of X1 =
1
Enter Value of X2 =
1
Enter Value of W1 =
1
Enter Value of W2 =
1
Net Input to the Output Neuron= 3
Output= 0.9051
```

Irrespective of the input, the output is always between -1 and 1

Generate AND/NOT function using McCulloch-Pitts neural net.

The McCulloch–Pitt neural network is considered to be the first neural network. The neurons are connected by directed weighted paths. McCulloch–Pitt neuron allows binary activation (ON or OFF), i.e., it either fires with an activation 1 or does not fire with an activation of 0.
If w > 0, then the connected path is said to be excitatory
If w < 0 ,then the connected path is said to be inhibitory.
Excitatory connections have positive weights and inhibitory connections have negative weights. Each neuron has a fixed threshold for firing. That is, if the net input to the neuron is greater than the threshold, it fires.
In the present case we implement AND-NOT function using McCulloch-Pitts Neural Network.
Binary inputs are taken, and the weights are initialized to 0
We have the following training sets and the target to implement the AND-NOT function

Training Sets		Target
X1	X2	t
0	0	0
0	1	0
1	0	1
1	1	0

The Python code is

```
w1 = 0
w2 = 0
print("For the ", 4, " inputs calculate the net input using yin = x1w1 + x2w2 ")

x1 = [0,0,1,1]
x2 = [0,1,0,1]
y = [0,0,1,0]
b = 0
print("Training Sets");
print("x1 = ",x1)
print("x2 = ",x2)
print("Target(y) :",y);

for i in range(0,4):
    w1 = w1 + x1[i] * y[i];
    w2 = w2 + x2[i] * y[i];
    b = b + y[i];
print("The updated weights are: ");
    print("W1new=",w1);
    print("w2new=",w2);
    print("bnew=",b);

print("The Final Weights are :")
print("w1new=",w1);
print("w2new=",w2);
print("bnew=",b);
```

The output of the programs is

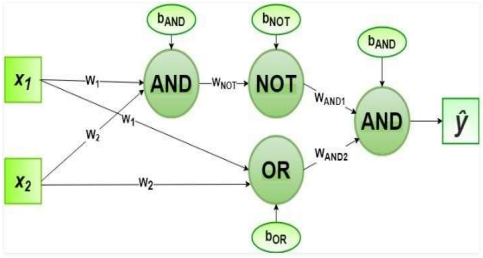
For the 4 inputs calculate the net input using yin = x1w1 + x2w2
Training Sets
x1 = [0, 0, 1, 1]
x2 = [0, 1, 0, 1]
Target(y) : [0, 0, 1, 0]
The updated weights are:
W1new= 0
w2new= 0
bnew= 0
The updated weights are:
W1new= 0
w2new= 0
bnew= 0
The updated weights are:
W1new= 1
w2new= 0
bnew= 1
The updated weights are:
W1new= 1
w2new= 0
bnew= 1
The Final Weights are :
w1new= 1
w2new= 0
bnew= 1

Generate XOR function using McCulloch-Pitts neural net.

We need to implement XOR function using McCulloch-Pitts Neural Network.
Binary inputs are taken, and the weights are initialized to suitable values.
We have the following training sets and the target to implement the EX-OR function

Training Sets		Target
X1	X2	t
0	0	0
0	1	1
1	0	1
1	1	0

As XOR function is not variable separable, we need some different strategy to implement this function.
Consider the diagram below



The weights are initialized as
$W_1 = 1, W_2 = 1,$
$W_{AND1} = 1, W_{AND2} = 1,$
$W_{NOT} = -1,$
$b_{AND} = -1.5, b_{OR} = -0.5,$
$b_{NOT} = 0.5$

The Python code for the given program is

```
# importing Python library
import numpy as np

# define Unit Step Function
def unitStep(v):
    if v >= 0:
        return 1
    else:
        return 0

# design Perceptron Model
def perceptronModel(x, w, b):
    v = np.dot(w, x) + b
    y = unitStep(v)
    return y

# NOT Logic Function
# wNOT = -1, bNOT = 0.5
def NOT_logicFunction(x):
    wNOT = -1
    bNOT = 0.5
```

```
    return perceptronModel(x, wNOT, bNOT)

# AND Logic Function
# here w1 = wAND1 = 1,
# w2 = wAND2 = 1, bAND = -1.5
def AND_logicFunction(x):
    w = np.array([1, 1])
    bAND = -1.5
    return perceptronModel(x, w, bAND)

# OR Logic Function
# w1 = 1, w2 = 1, bOR = -0.5
def OR_logicFunction(x):
    w = np.array([1, 1])
    bOR = -0.5
    return perceptronModel(x, w, bOR)

# XOR Logic Function
# with AND, OR and NOT
# function calls in sequence
def XOR_logicFunction(x):
    y1 = AND_logicFunction(x)
    y2  = OR_logicFunction(x)
    y3 = NOT_logicFunction(y1)
    final_x = np.array([y2, y3])
    finalOutput = AND_logicFunction(final_x)
    return finalOutput

# testing the Perceptron Model
test1 = np.array([0, 0])
test2 = np.array([0, 1])
test3 = np.array([1, 0])
test4 = np.array([1, 1])

print("XOR({}, {}) = {}".format(0, 0, XOR_logicFunction(test1)))
print("XOR({}, {}) = {}".format(0, 1, XOR_logicFunction(test2)))
print("XOR({}, {}) = {}".format(1, 0, XOR_logicFunction(test3)))
print("XOR({}, {}) = {}".format(1, 1, XOR_logicFunction(test4)))
```

We get the following output

```
XOR(0, 0) = 0
XOR(0, 1) = 1
XOR(1, 0) = 1
XOR(1, 1) = 0
```

Write a program to implement Hebb's rule

Hebbian Learning Rule, also known as Hebb Learning Rule, was proposed by Donald O Hebb. It is one of the first and also easiest learning rules in the neural network. It is used for pattern classification.
The Hebbian Learning Rule is a learning rule that specifies how much the weight of the connection between two units should be increased or decreased in proportion to the product of their activation. The rule builds on Hebb's 1949 learning rule which states that the connections between two neurons might be strengthened if the neurons fire simultaneously. The Hebbian Rule works well as long as all the input patterns are orthogonal or uncorrelated. The requirement of orthogonality places serious limitations on the Hebbian Learning Rule. For the present case we consider the example of OR function, using Bipolar input we have the following relations between the training sets and the target

Training Sets		Target
X1	X2	t
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

The Python code for the given case is

```
w1 = 0
w2 = 0
b = 0

x1 = [-1, -1, 1, 1]
x2 = [-1, 1, -1, 1]
y = [-1, 1, 1, 1]

def train_perceptron(x1, x2, y):
    global w1, w2, b
    for i in range(len(x1)):
        w1 += x1[i] * y[i]
        w2 += x2[i] * y[i]
        b += y[i]
        print("Epoch {}: w1_new={}, w2_new={}, b_new={}".format(i+1, w1, w2, b))

# Training with OR examples
print("Training with OR examples:")
train_perceptron(x1, x2, y)

# Display the final weights and bias
print("\nThe Final Weights are:")
print("w1_new =", w1)
print("w2_new =", w2)
print("b_new =", b)
```

The output is as follows

Training with OR examples:
Epoch 1: w1_new=1, w2_new=1, b_new=-1
Epoch 2: w1_new=0, w2_new=2, b_new=0
Epoch 3: w1_new=1, w2_new=1, b_new=1
Epoch 4: w1_new=2, w2_new=2, b_new=2

The Final Weights are:
w1_new = 2
w2_new = 2
b_new = 2

Implementing the Delta rule

The Delta rule in machine learning and neural network environments is a specific type of Backpropagation that helps to refine connectionist ML/AI networks, making connections between inputs and outputs with layers of artificial neurons.

The Delta rule is also known as the Delta learning rule, in general, Backpropagation has to do with recalculating input weights for artificial neurons using a gradient method. Delta learning does this using the difference between a target activation and an actual obtained activation. Using a linear activation function, network connections are adjusted.

Another way to explain the Delta rule is that it uses an error function to perform gradient descent learning.

A tutorial on the Delta rule explains that essentially in comparing an actual output with a targeted output, the technology tries to find a match. If there is not a match, the program makes changes. The actual implementation of the Delta rule is going to vary according to the network and its composition, but by employing a linear activation function, the Delta rule can be useful in refining some types of neural network systems with particular flavour of Backpropagation.

For the present case we take the example of OR function using Bipolar inputs

Training Sets		Target
X1	X2	t
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

The Python code is as follows

```
import numpy as np

# Input data for OR function
X = np.array([
      [0, 0],
      [0, 1],
      [1, 0],
      [1, 1]
])

# Target values for OR function
Y = np.array([0, 1, 1, 1])

# Initialize weights and bias
weights = np.random.rand(2)    # Initialize with random values
bias = np.random.rand(1)

# Learning rate
learning_rate = 0.1

# Number of epochs
```

```
epochs = 10

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def delta_rule_train(X, Y, weights, bias, learning_rate, epochs):
    for epoch in range(epochs):
        for i in range(len(X)):
            # Forward pass
            output = sigmoid(np.dot(X[i], weights) + bias)

            # Compute error
            error = Y[i] - output

            # Update weights and bias using the Delta rule
            weights += learning_rate * error * X[i]
            bias += learning_rate * error

        # Display the error for every 100 epochs
        if epoch % 100 == 0:
            total_error = np.mean(np.square(Y - sigmoid(np.dot(X, weights) + bias)))
            print("Epoch {}: Total Error: {}".format(epoch, total_error))

    return weights, bias

# Train the perceptron using the Delta rule
trained_weights, trained_bias = delta_rule_train(X, Y, weights, bias, learning_rate, epochs)

# Display the final weights and bias rounded to two decimal places
print("\nThe Final Weights are:")
print("w1_new =", round(trained_weights[0], 2))
print("w2_new =", round(trained_weights[1], 2))
print("b_new =", round(trained_bias[0], 2))
```

We get the following output

```
Epoch 0: Total Error: 0.15009771759263787

The Final Weights are:
w1_new = 0.97
w2_new = 0.67
b_new = 0.43
```

Implementing Back Propagation Algorithm

Backpropagation, short for "backward propagation of errors," is a supervised learning algorithm used for training artificial neural networks. It is a widely used optimization algorithm that helps adjust the weights of the neural network to minimize the difference between the predicted output and the actual output.

Explanation of backpropagation:

1. Forward Pass:
   - The input data is fed forward through the network, layer by layer, to generate the predicted output.
   - Each neuron in the network applies an activation function to its input, producing an output that becomes the input for the next layer.

2. Compute Loss:
   - The predicted output is compared to the actual target output, and the error (or loss) is calculated. This measures how far off the network's predictions are from the true values.

3. Backward Pass (Backpropagation):
   - The algorithm then works backward through the network to compute the gradients of the loss with respect to the weights.
   - The gradients represent how much the loss would increase or decrease if the weights were adjusted.

4. Weight Update:
   - The weights are then updated using an optimization algorithm, such as gradient descent, to minimize the loss. The learning rate determines the size of the steps taken during this update.

5. Iteration:
   - Steps 1-4 are repeated for multiple iterations (epochs) until the network has learned the underlying patterns in the training data.

The key idea behind backpropagation is to use the chain rule from calculus to compute the gradient of the loss with respect to the weights. This allows the network to adjust its weights in a way that reduces the error in predicting the target values.

The backpropagation algorithm makes training deep neural networks feasible by efficiently computing the gradients for all the weights in the network. It's a crucial component in the training of neural networks for a wide range of tasks, including image classification, natural language processing, and many others.

The Python code for the given case is

```
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(x):
    return x    (1 - x)

def initialize_weights(input_size, hidden_size, output_size):
    np.random.seed(42)
    weights_input_hidden = np.random.rand(input_size, hidden_size)
    weights_hidden_output = np.random.rand(hidden_size, output_size)
    return weights_input_hidden, weights_hidden_output

def forward_pass(inputs, weights_input_hidden, weights_hidden_output):
    hidden_layer_input = np.dot(inputs, weights_input_hidden)
    hidden_layer_output = sigmoid(hidden_layer_input)

    output_layer_input = np.dot(hidden_layer_output, weights_hidden_output)
    output_layer_output = sigmoid(output_layer_input)

    return hidden_layer_output, output_layer_output

def backward_pass(inputs, targets, hidden_layer_output, output_layer_output,
weights_hidden_output):
    output_error = targets - output_layer_output
    output_delta = output_error     sigmoid_derivative(output_layer_output)

    hidden_layer_error = output_delta.dot(weights_hidden_output.T)
    hidden_layer_delta = hidden_layer_error     sigmoid_derivative(hidden_layer_output)

    return output_delta, hidden_layer_delta

def update_weights(inputs, hidden_layer_output, output_delta, hidden_layer_delta,
weights_input_hidden, weights_hidden_output, learning_rate=0.1):
    weights_hidden_output += hidden_layer_output.T.dot(output_delta)    learning_rate
    weights_input_hidden += inputs.T.dot(hidden_layer_delta)    learning_rate

def train_neural_network(inputs, targets, epochs=10000):
    input_size = inputs.shape[1]
    hidden_size = 4    # You can adjust the number of hidden layer neurons
    output_size = 1

    weights_input_hidden, weights_hidden_output = initialize_weights(input_size, hidden_size,
output_size)

    for epoch in range(epochs):
        # Forward pass
        hidden_layer_output, output_layer_output = forward_pass(inputs, weights_input_hidden,
weights_hidden_output)
```

```
            # Backward pass
            output_delta, hidden_layer_delta = backward_pass(inputs, targets, hidden_layer_output,
output_layer_output, weights_hidden_output)

            # Update weights
            update_weights(inputs, hidden_layer_output, output_delta, hidden_layer_delta,
weights_input_hidden, weights_hidden_output)

    return weights_input_hidden, weights_hidden_output

def predict(inputs, weights_input_hidden, weights_hidden_output):
    _, output = forward_pass(inputs, weights_input_hidden, weights_hidden_output)
    return np.round(output)

# Example usage:
# Training data for the OR function
inputs = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
targets = np.array([[0], [1], [1], [1]])

# Train the neural network
trained_weights_input_hidden, trained_weights_hidden_output = train_neural_network(inputs,
targets)

# Test the trained neural network
for input_pattern, target in zip(inputs, targets):
    prediction = predict(input_pattern, trained_weights_input_hidden,
trained_weights_hidden_output)
    print(f"{input_pattern} -> {prediction} (target: {target})")
```

We get the following output

```
[0 0] -> [0.] (target: [0])
[0 1] -> [1.] (target: [1])
[1 0] -> [1.] (target: [1])
[1 1] -> [1.] (target: [1])
```

Implementing Error Back Propagation Algorithm

"Error backpropagation" is another term for the backpropagation algorithm in the context of neural networks. It refers to the process of propagating the error backward through the network during the training phase.

In the context of neural networks, the term "error" typically refers to the difference between the predicted output of the network and the actual target output. The goal of the training process is to minimize this error, and the backpropagation algorithm is a key tool for achieving that.

Error backpropagation works as follows

1. Forward Pass:
   - Input data is fed forward through the network to generate the predicted output.
   - Each layer applies an activation function to its input, producing an output.

2. Compute Error:
   - The predicted output is compared to the actual target output, and the error (loss) is calculated.

3. Backward Pass (Backpropagation):
   - The algorithm works backward through the network to compute the gradients of the loss with respect to the weights.
   - It uses the chain rule of calculus to efficiently calculate how much each weight contributed to the error.

4. Weight Update:
   - The weights are updated using an optimization algorithm (e.g., gradient descent) based on the calculated gradients. This step aims to reduce the error by adjusting the weights in a direction that minimizes the loss.

5. Iteration:
   - Steps 1-4 are repeated for multiple iterations (epochs) until the network has learned to make accurate predictions.

The term "backpropagation" emphasizes the backward flow of information during the training process. The error calculated at the output layer is propagated backward through the network, and the weights are updated accordingly. This iterative process allows the neural network to learn from its mistakes and improve its performance over time.

Python Code

```
import numpy as np

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(x):
    return x * (1 - x)

def initialize_weights(input_size, hidden_size, output_size):
    np.random.seed(42)
    weights_input_hidden = np.random.rand(input_size, hidden_size)
    weights_hidden_output = np.random.rand(hidden_size, output_size)
    return weights_input_hidden, weights_hidden_output

def forward_pass(inputs, weights_input_hidden, weights_hidden_output):
    hidden_layer_input = np.dot(inputs, weights_input_hidden)
    hidden_layer_output = sigmoid(hidden_layer_input)

    output_layer_input = np.dot(hidden_layer_output, weights_hidden_output)
    output_layer_output = sigmoid(output_layer_input)

    return hidden_layer_output, output_layer_output

def calculate_error(targets, output_layer_output):
    return targets - output_layer_output

def backward_pass(inputs, hidden_layer_output, output_layer_output, error,
weights_hidden_output):
    output_delta = error * sigmoid_derivative(output_layer_output)

    hidden_layer_error = output_delta.dot(weights_hidden_output.T)
    hidden_layer_delta = hidden_layer_error * sigmoid_derivative(hidden_layer_output)

    return output_delta, hidden_layer_delta

def update_weights(inputs, hidden_layer_output, output_delta, hidden_layer_delta,
weights_input_hidden, weights_hidden_output, learning_rate=0.1):
    weights_hidden_output += hidden_layer_output.T.dot(output_delta) * learning_rate
    weights_input_hidden += inputs.T.dot(hidden_layer_delta) * learning_rate

def train_neural_network(inputs, targets, epochs=10000):
    input_size = inputs.shape[1]
    hidden_size = 4    # You can adjust the number of hidden layer neurons
    output_size = 1

    weights_input_hidden, weights_hidden_output = initialize_weights(input_size,
hidden_size, output_size)

    for epoch in range(epochs):
        # Forward pass
        hidden_layer_output, output_layer_output = forward_pass(inputs,
weights_input_hidden, weights_hidden_output)
```

```
            # Calculate error
            error = calculate_error(targets, output_layer_output)

            # Backward pass
            output_delta, hidden_layer_delta = backward_pass(inputs, hidden_layer_output,
output_layer_output, error, weights_hidden_output)

            # Update weights
            update_weights(inputs, hidden_layer_output, output_delta, hidden_layer_delta,
weights_input_hidden, weights_hidden_output)

    return weights_input_hidden, weights_hidden_output

def predict(inputs, weights_input_hidden, weights_hidden_output):
    _, output = forward_pass(inputs, weights_input_hidden, weights_hidden_output)
    return np.round(output)

# Example usage:
# Training data for the OR function
inputs = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
targets = np.array([[0], [1], [1], [1]])

# Train the neural network
trained_weights_input_hidden, trained_weights_hidden_output = train_neural_network(inputs,
targets)

# Test the trained neural network
for input_pattern, target in zip(inputs, targets):
    prediction = predict(input_pattern, trained_weights_input_hidden,
trained_weights_hidden_output)
    print(f"{input_pattern} -> {prediction} (target: {target})")
```

We get the following output

```
[0 0] -> [0.] (target: [0])
[0 1] -> [1.] (target: [1])
[1 0] -> [1.] (target: [1])
[1 1] -> [1.] (target: [1])
```

Implementing Hopfield Network

A Hopfield network is a type of recurrent artificial neural network, primarily used for associative memory and pattern recognition. It was introduced by John Hopfield in 1982. Hopfield networks are single-layer networks with symmetric connections, meaning that the weights are the same for connections going from one neuron to another and from the second neuron back to the first.

Key characteristics of Hopfield networks include:

1. Recurrence: Hopfield networks are fully connected and recurrent, meaning each neuron is connected to every other neuron, including itself. The recurrent connections allow the network to store and retrieve patterns.

2. Symmetric Weights: The connection weights in a Hopfield network are symmetric. If there is a connection from neuron i to neuron j, there is also a connection from neuron j to neuron i, and the weights are the same in both directions.

3. Binary Activation: Neurons in a Hopfield network typically use binary activation, where the states are either +1 or -1. The network operates in discrete time steps.

4. Energy Function: The network is designed to minimize an energy function during the learning process. The energy function is associated with the patterns the network is trained on.

Hopfield networks have applications in content-addressable memory and pattern recognition. They can be trained to store and retrieve patterns, and given a partial or noisy input, they can reconstruct the closest stored pattern. However, Hopfield networks have limitations, such as being prone to spurious patterns and having limitations in terms of storage capacity.

The update rule for a Hopfield network is typically based on the Hebbian learning rule, which adjusts weights based on the correlation between the activities of connected neurons.

While Hopfield networks have been influential in the development of neural network concepts, more advanced and scalable models, such as feedforward neural networks and recurrent neural networks, are commonly used for modern applications in machine learning and artificial intelligence.

The Python code for the given case is

```
import numpy as np
```

```
class HopfieldNetwork:
    def __init (self, size):
        self.size = size
        self.weights = np.zeros((size, size))

    def train(self, patterns):
        for pattern in patterns:
            pattern = np.reshape(pattern, (self.size, 1))
            self.weights += np.outer(pattern, pattern)
            np.fill_diagonal(self.weights, 0)

    def predict(self, input_pattern, max_iter=100):
        input_pattern = np.reshape(input_pattern, (self.size, 1))

        for _ in range(max_iter):
            output_pattern = np.sign(np.dot(self.weights, input_pattern))
            if np.array_equal(input_pattern, output_pattern):
                return output_pattern.flatten()
            input_pattern = output_pattern

        raise RuntimeError("Hopfield Network did not converge within the maximum number of
iterations.")

# Example usage:
# Define patterns for training
pattern1 = np.array([1, -1, 1, -1])
pattern2 = np.array([-1, -1, -1, 1])
patterns = [pattern1, pattern2]

# Create a Hopfield Network and train it
hopfield_net = HopfieldNetwork(size=len(pattern1))
hopfield_net.train(patterns)

# Test the Hopfield Network with a noisy input pattern
noisy_pattern = np.array([1, 1, 1, 1])
predicted_pattern = hopfield_net.predict(noisy_pattern)

# Display results
print("Noisy Pattern:", noisy_pattern)
print("Predicted Pattern:", predicted_pattern)
```

We get the following output

```
Noisy Pattern: [1 1 1 1]
Predicted Pattern: [ 1.  0.  1. -1.]
```

Implementing Radial Basis function

A Radial Basis Function (RBF) is a real-valued function whose output depends on the distance between the input and a fixed point in space, often referred to as the center. RBFs are commonly used in various fields, including machine learning, mathematics, and signal processing. In the context of machine learning, RBFs are often used as activation functions in certain types of neural networks or as kernels in support vector machines (SVMs).

In the context of neural networks, a Radial Basis Function Network (RBFN) is a type of artificial neural network that uses radial basis functions as activation functions.

The architecture typically consists of three layers:

1. Input Layer: Neurons in the input layer represent the input features.

2. Hidden Layer with Radial Basis Functions: Neurons in the hidden layer apply radial basis functions to transform the input. Each neuron in the hidden layer is associated with a center, and its output is a function of the distance between the input and the center.

3. Output Layer: Neurons in the output layer combine the outputs of the hidden layer to produce the final network output.

The RBFN learns the centers and widths of the radial basis functions during the training process.

In SVMs, RBFs are commonly used as kernels in the kernelized version of the algorithm. The RBF kernel measures the similarity (or distance) between data points in the input space.

The use of radial basis functions allows these models to capture complex, non-linear relationships in the data.

The Python code for the given case is

```
importnumpy as np
fromscipy.spatial.distance import cdist
```

```
classRadialBasisFunctionNetwork:
def __init__(self, input_size, hidden_size, output_size):
self.input_size = input_size
self.hidden_size = hidden_size
self.output_size = output_size
self.centers = None
self.width = None
self.weights_hidden_output = np.random.rand(hidden_size, output_size)

definitialize_centers(self, inputs):
        # Use k-means clustering to initialize centers
indices = np.random.choice(len(inputs), self.hidden_size, replace=False)
self.centers = inputs[indices]

defcalculate_width(self):
        # Set the width as the maximum distance between centers
distances = cdist(self.centers, self.centers, 'euclidean')
self.width = np.max(distances) / np.sqrt(2 * self.hidden_size)

defradial_basis_function(self, x, c, width):
returnnp.exp(-np.linalg.norm(x - c) / (2 * width**2))

def train(self, inputs, targets, learning_rate=0.01, epochs=1000):
self.initialize_centers(inputs)
self.calculate_width()

for epoch in range(epochs):
for i in range(len(inputs)):
                # Calculate activations for hidden layer (RBF layer)
hidden_layer_output = np.array([self.radial_basis_function(inputs[i], c, self.width) for c in
self.centers])

                # Calculate output layer (linear)
output_layer_output = np.dot(hidden_layer_output, self.weights_hidden_output)

                # Update weights using gradient descent
self.weights_hidden_output += learning_rate * np.outer(hidden_layer_output, targets[i] -
output_layer_output)

def predict(self, inputs):
hidden_layer_output = np.array([self.radial_basis_function(inputs, c, self.width) for c in
self.centers])
output_layer_output = np.dot(hidden_layer_output, self.weights_hidden_output)
returnoutput_layer_output

# Example usage:
# Define input patterns and corresponding targets
inputs = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
targets = np.array([[0], [1], [1], [0]])

# Create an RBF Network and train it
```

```
rbf_net = RadialBasisFunctionNetwork(input_size=2, hidden_size=4, output_size=1)
rbf_net.train(inputs, targets)

# Test the RBF Network
forinput_pattern, target in zip(inputs, targets):
predicted_output = rbf_net.predict(input_pattern)
print(f"Input: {input_pattern}, Target: {target}, Predicted Output: {predicted_output}")
```

We get the following output

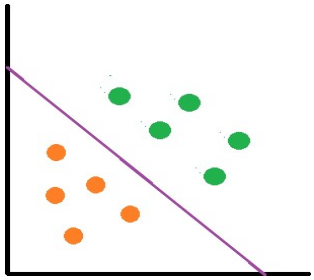
```
Input: [0 0], Target: [0], Predicted Output: [0.0012516]
Input: [0 1], Target: [1], Predicted Output: [0.99879928]
Input: [1 0], Target: [1], Predicted Output: [0.99879736]
Input: [1 1], Target: [0], Predicted Output: [0.00115243]
```

Write a program for Linear separation

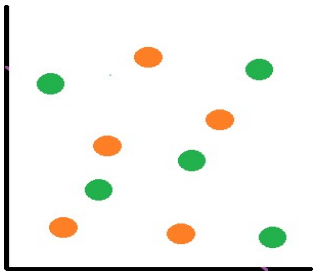
Linear separable means that there is a hyperplane, which splits the input data into two half-spaces such that all points of the first class should be in one half-space and other points of the second class should be in the other half-space.

In 2-D, it means that there is a line, which separates points of one class from points of the other class.

For example**:** In the following image, if orange circles represent points from one class and green circles represent points from the other class, then these points are linearly separable.



In the following image, the two classes represented by orange and green circles cannot be separated by a line



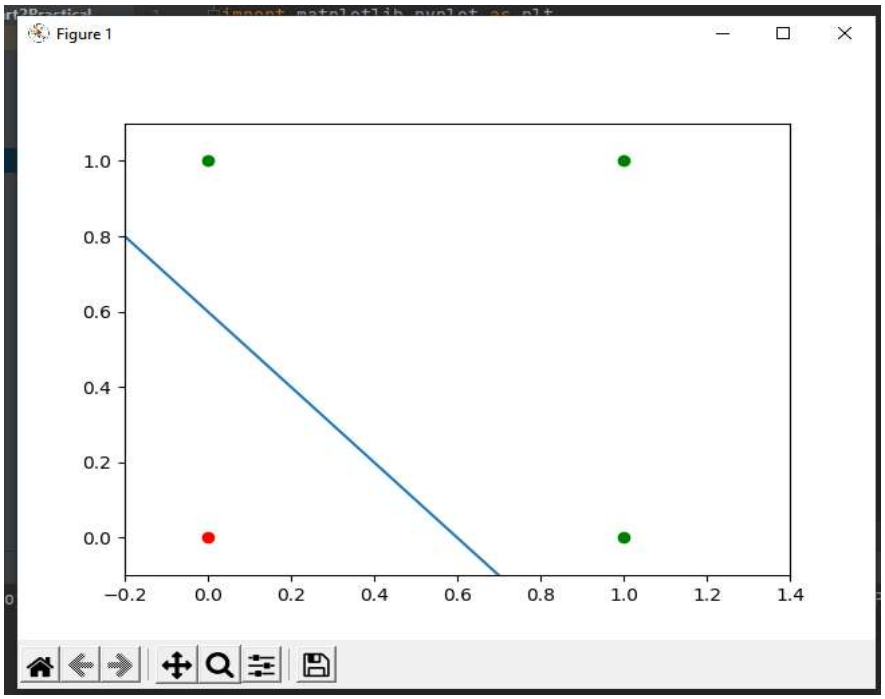
Linear separability is an important concept in neural networks.
In ANN a decision line is drawn to separate positive and negative responses. The decision line may also be called as the decision-making Line or decision-support Line or linear-separable line. The necessity of the linear separability concept was felt to clarify classify the patterns based upon their output responses

The Logic functions AND, OR etc are linearly separable, while XOR and EX-NOR are not linearly separable.

The following program shows the linear separability of OR functions.

```
import matplotlib.pyplot as plt
import numpy as np


fig, ax = plt.subplots()
xmin, xmax = -0.2, 1.4
X = np.arange(xmin, xmax, 0.1)
ax.scatter(0, 0, color="r")
ax.scatter(0, 1, color="g")
ax.scatter(1, 0, color="g")
ax.scatter(1, 1, color="g")
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m = -1
ax.plot(X, m * X + 0.6, label="decision boundary")
plt.show()
#plt.plot()
```

We get the following plot



Green dots show output as 1 and red shows output as 0

Membership and Identity Operators is, is not

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y.

Membership Operators (in and not in)

```
# Lists
fruits = ["apple", "banana", "cherry"]

# Check if an element is in the list
if "banana" in fruits:
    print("banana is in the list")

# Check if an element is not in the list
if "orange" not in fruits:
    print("orange is not in the list")

# Strings
text = "Hello, world"

# Check if a substring is in the string
if "world" in text:
    print("The word 'world' is in the string")
```

Identity Operators (is and is not)

```
# Numbers
x = 10
y = 10

# Check if two variables reference the same object
if x is y:
    print("x and y are the same object")

# Check if two variables do not reference the same object
if x is not "Hello":
    print("x is not 'Hello'")

# Lists
list1 = [1, 2, 3]
list2 = [1, 2, 3]

# Check if two lists are not the same object (even though they have the same contents)
if list1 is not list2:
    print("list1 and list2 are not the same object")
```

Find ratios using fuzzy logic

Ratio

We used the ratio function above to calculate the Levenshtein distance similarity ratio between the two strings (sequences).

Partial Ratio

FuzzyWuzzy also has more powerful functions to help with matching strings in more complex situations. This function allows us to perform substring matching. This works by taking the shortest string and matching it with all substrings that are of the same length.

Token Sort Ratio

FuzzyWuzzy also has token functions that tokenize the strings, change capitals to lowercase, and remove punctuation. This function sorts the strings alphabetically and then joins them together. Then, the fuzz.ratio() is calculated. This can come in handy when the strings we are comparing are the same in spelling but are not in the same order.

Token Set Ratio

This function is similar to the token sort ratio, except it takes out the common tokens before calculating the fuzz ration between the new strings. This function is the most helpful when applied to a set of strings with a significant difference in lengths.

```
# Python code showing all the ratios together,

# make sure you have installed fuzzywuzzy module


from fuzzywuzzy import fuzz

from fuzzywuzzy import process


s1 = "I like softcomputing"

s2 = "I like hardcomputing"

print ("FuzzyWuzzy Ratio: ", fuzz.ratio(s1, s2))

print ("FuzzyWuzzy PartialRatio: ", fuzz.partial_ratio(s1, s2))

print ("FuzzyWuzzy TokenSortRatio: ", fuzz.token_sort_ratio(s1, s2))

print ("FuzzyWuzzy TokenSetRatio: ", fuzz.token_set_ratio(s1, s2))

print ("FuzzyWuzzy WRatio: ", fuzz.WRatio(s1, s2),'\n\n')
```

We get the following output

FuzzyWuzzy Ratio: 80
FuzzyWuzzy PartialRatio: 80
FuzzyWuzzy TokenSortRatio: 45
FuzzyWuzzy TokenSetRatio: 80
FuzzyWuzzy WRatio: 80

Smt . Durgadevi Sharma Charitable Trust's

Chandrabhan Sharma College of Arts, Commerce & Science (Autonomous)

Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

Subject Name : **DATA SCIENCE**

Name : **PRATIK JAGDISHCHANDRA MAURYA**

Seat No : **1310226**

Teaching Faculty : **Mr. Sandeep Vishwakarma**

DEPARTMENT OF INFORMATION TECHNOLOGY

CHANDRABHAN SHARMA COLLEGE OF ARTS,

COMMERCE & SCIENCE

NAAC RE-ACCREDITED 'A' Grade (CGPA 3.10)

(Autonomous)

MUMBAI, 400076

MAHARASHTRA

2024-2025

Smt . Durgadevi Sharma Charitable Trust's
Chandrabhan Sharma College
of Arts, Commerce & Science
(Autonomous)
Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that Mr./~~Miss~~ **PRATIK JAGDISHCHANDRA MAURYA** having Exam SeatNo. **1310226** of M.Sc.IT **(Semester I)** has complete the Practical work in the subject of "**Data Science**" during the **Academic year 2024-25** under the guidance of **Mr. Sandeep Vishwakarma** being the Partial requirement for The fulfilment of the curriculum of Degree of Master of Science in Information Technology, University of Mumbai.

Internal Examiner **Co-Ordinator**

Date: **College Seal**

External Examiner

INDEX

	E	Building a diagram for scheduling of jobs		
	F	Connecting other Data Sources		
Practical 5		**Assessing Data**		
Practical 6		**Processing Data**		
Practical 7		**Transforming Data**		
Practical 8		**Organizing Data**		
	A	Horizontal Style		
	B	Vertical Style		
	C	Island Style		
	D	Secure Vault Style		
Practical 9		**Reporting Data**		
	A	Create a network routing diagram		
	B	Directed Acyclic Graph		
	C	Graphics		
Practical 10		**Working with Power BI**		
	A	Importing data from Excel		
	B	Importing data from OData Feed		
	C	Data Visualization with Power BI		

Practical 1

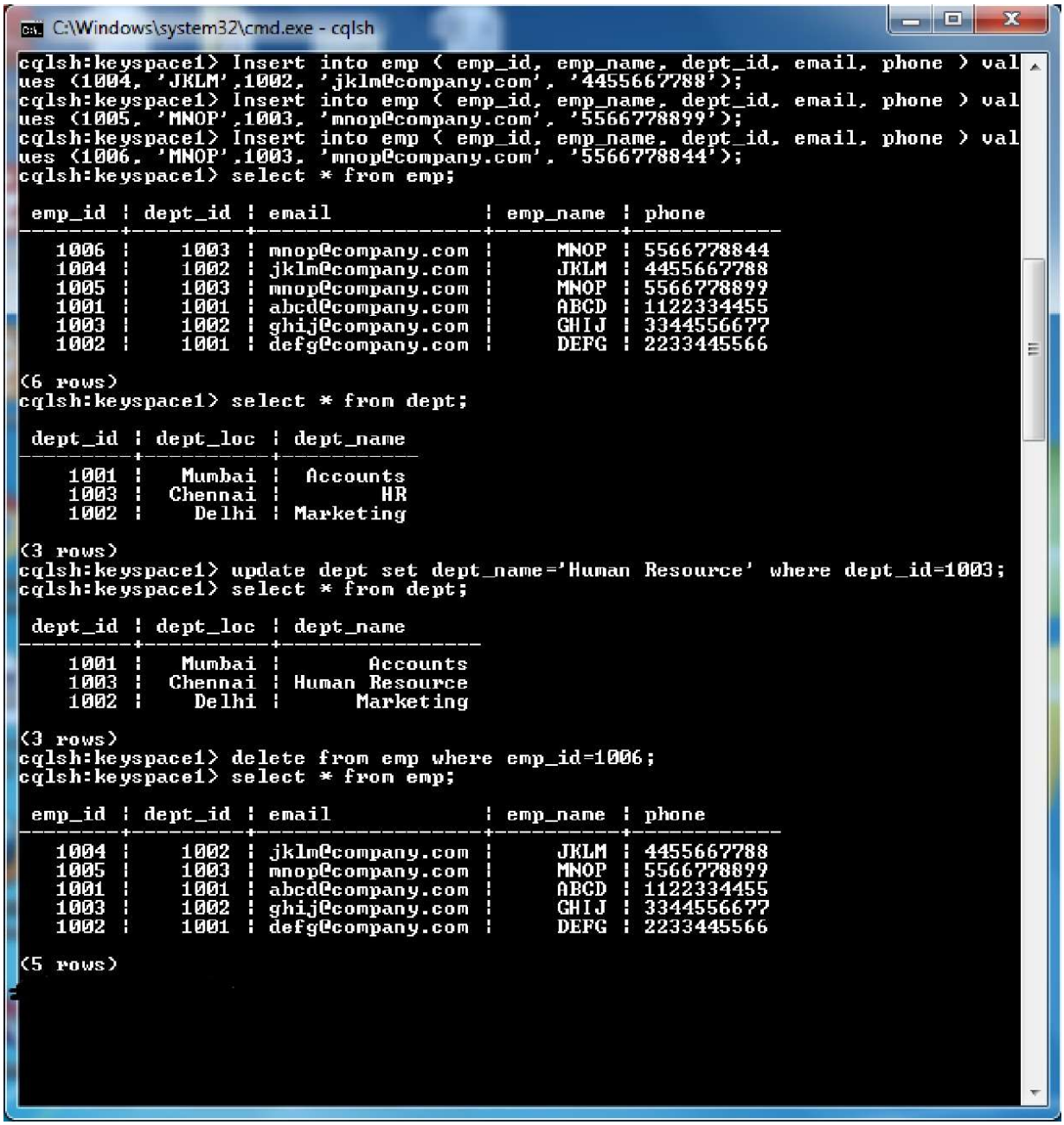
Creating Data Model using Cassandra

Cassandra Data Model

Code:

```
Create keyspace keyspace1 with replication = {'class':'SimpleStrategy','replication_factor': 3};

Use keyspace1;

Create table dept ( dept_id int PRIMARY KEY, dept_name text, dept_loc text);

Create table emp ( emp_id int PRIMARY KEY, emp_name text, dept_id int, email text, phone text );

Insert into dept (dept_id, dept_name, dept_loc) values (1001, 'Accounts', 'Mumbai');

Insert into dept (dept_id, dept_name, dept_loc) values (1002, 'Marketing', 'Delhi');

Insert into dept (dept_id, dept_name, dept_loc) values (1003, 'HR', 'Chennai');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1001, 'ABCD',1001,
'abcd@company.com', '1122334455');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1002, 'DEFG',1001,
'defg@company.com', '2233445566');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1003, 'GHIJ',1002,
'ghij@company.com', '3344556677');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1004, 'JKLM',1002,
'jklm@company.com', '4455667788');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1005, 'MNOP',1003,
'mnop@company.com', '5566778899');

Insert into emp ( emp_id, emp_name, dept_id, email, phone ) values (1006, 'MNOP',1003,
'mnop@company.com', '5566778844');

select * from emp;

select * from dept;

update dept set dept_name='Human Resource' where dept_id=1003;

delete from emp where emp_id=1006;

select * from emp;
```

Output:


C:\Windows\system32\cmd.exe - cqlsh
cqlsh:keyspace1> Insert into emp (emp_id, emp_name, dept_id, email, phone) val
ues (1004, 'JKLM',1002, 'jklm@company.com', '4455667788');
cqlsh:keyspace1> Insert into emp (emp_id, emp_name, dept_id, email, phone) val
ues (1005, 'MNOP',1003, 'mnop@company.com', '5566778899');
cqlsh:keyspace1> Insert into emp (emp_id, emp_name, dept_id, email, phone) val
ues (1006, 'MNOP',1003, 'mnop@company.com', '5566778844');
cqlsh:keyspace1> select * from emp;
emp_id | dept_id | email | emp_name | phone
1006 | 1003 | mnop@company.com | MNOP | 5566778844
1004 | 1002 | jklm@company.com | JKLM | 4455667788
1005 | 1003 | mnop@company.com | MNOP | 5566778899
1001 | 1001 | abcd@company.com | ABCD | 1122334455
1003 | 1002 | ghij@company.com | GHIJ | 3344556677
1002 | 1001 | defg@company.com | DEFG | 2233445566
(6 rows)
cqlsh:keyspace1> select * from dept;
dept_id | dept_loc | dept_name
1001 | Mumbai | Accounts
1003 | Chennai | HR
1002 | Delhi | Marketing
(3 rows)
cqlsh:keyspace1> update dept set dept_name='Human Resource' where dept_id=1003;
cqlsh:keyspace1> select * from dept;
dept_id | dept_loc | dept_name
1001 | Mumbai | Accounts
1003 | Chennai | Human Resource
1002 | Delhi | Marketing
(3 rows)
cqlsh:keyspace1> delete from emp where emp_id=1006;
cqlsh:keyspace1> select * from emp;
emp_id | dept_id | email | emp_name | phone
1004 | 1002 | jklm@company.com | JKLM | 4455667788
1005 | 1003 | mnop@company.com | MNOP | 5566778899
1001 | 1001 | abcd@company.com | ABCD | 1122334455
1003 | 1002 | ghij@company.com | GHIJ | 3344556677
1002 | 1001 | defg@company.com | DEFG | 2233445566
(5 rows)

Practical 2

Conversion from different formats to HORUS formats

A) Csv to Horus

Code:

```
import pandas as pd
# Input Agreement ============================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/Country_Code.csv'
InputData=pd.read_csv(sInputFileName,encoding="latin-1")
print('Input Data Values ===================================')
print(InputData)
print('=====================================================')
# Processing Rules ===========================================
ProcessData=InputData
# Remove columns ISO-2-Code and ISO-3-CODE
ProcessData.drop('ISO-2-CODE', axis=1,inplace=True)
ProcessData.drop('ISO-3-Code',  axis=1,inplace=True)
# Rename Country and ISO-M49
ProcessData.rename(columns={'Country': 'CountryName'}, inplace=True)
ProcessData.rename(columns={'ISO-M49': 'CountryNumber'}, inplace=True)
# Set new Index
ProcessData.set_index('CountryNumber', inplace=True)
# Sort data by CurrencyNumber
ProcessData.sort_values('CountryName', axis=0, ascending=False, inplace=True)
print('Process Data Values ================================')
print(ProcessData)
print('=====================================================')
# Output Agreement ===========================================
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-CSV-Country.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('CSV to HORUS - Done')
# Utility done
```

Output:

```
In [93]: runfile('H:/VKHCG/05-DS/9999-Data/CSV2HORUS.py', wdir='H:/VKHCG/05-DS/9999-Data')


Input Data Values ==================================
                         Country ISO-2-CODE ISO-3-Code  ISO-M49
0                    Afghanistan         AF        AFG        4
1                  Aland Islands         AX        ALA      248
2                        Albania         AL        ALB        8
3                        Algeria         DZ        DZA       12
4                 American Samoa         AS        ASM       16
5                        Andorra         AD        AND       20
6                         Angola         AO        AGO       24
7                       Anguilla         AI        AIA      660
8                     Antarctica         AQ        ATA       10
9            Antigua and Barbuda         AG        ATG       28
10                     Argentina         AR        ARG       32
11                       Armenia         AM        ARM       51
12                         Aruba         AW        ABW      533
13                     Australia         AU        AUS       36
14                       Austria         AT        AUT       40
15                    Azerbaijan         AZ        AZE       31
16                       Bahamas         BS        BHS       44
17                       Bahrain         BH        BHR       48
18                    Bangladesh         BD        BGD       50
19                      Barbados         BB        BRB       52
20                       Belarus         BY        BLR      112
21                       Belgium         BE        BEL       56
22                        Belize         BZ        BLZ       84
23                         Benin         BJ        BEN      204
24                       Bermuda         BM        BMU       60
25                        Bhutan         BT        BTN       64
26                       Bolivia         BO        BOL       68
27     Bosnia and Herzegovina            BA        BIH       70
28                      Botswana         BW        BWA       72
29                  Bouvet Island        BV        BVT       74
..                            ...        ...        ...      ...
217                  Tajikistan          TJ        TJK      762
218  Tanzania, United Republic of        TZ        TZA      834
219                    Thailand          TH        THA      764
220                 Timor-Leste          TL        TLS      626
221                        Togo          TG        TGO      768
222                     Tokelau          TK        TKL      772
223                       Tonga          TO        TON      776
224         Trinidad and Tobago          TT        TTO      780
225                     Tunisia          TN        TUN      788
226                      Turkey          TR        TUR      792
227                Turkmenistan          TM        TKM      795
228     Turks and Caicos Islands         TC        TCA      796
229                      Tuvalu          TV        TUV      798
230                      Uganda          UG        UGA      800
231                     Ukraine          UA        UKR      804
232        United Arab Emirates          AE        ARE      784
233              United Kingdom          GB        GBR      826
234       United States of America       US        USA      840
235  US Minor Outlying Islands           UM        UMI      581
236                     Uruguay          UY        URY      858
237                  Uzbekistan          UZ        UZB      860
238                     Vanuatu          VU        VUT      548
239  Venezuelaÿ(Bolivarian Republic)     VE        VEN      862
240                    Viet Nam          VN        VNM      704
241             Virgin Islands, US       VI        VIR      850
242       Wallis and Futuna Islands      WF        WLF      876
243                  Western Sahara       EH        ESH      732
244                       Yemen          YE        YEM      887
245                      Zambia          ZM        ZMB      894
246                    Zimbabwe          ZW        ZWE      716

[247 rows x 4 columns]
====================================================
Process Data Values ==================================
                              CountryName
CountryNumber
716                             Zimbabwe
894                              Zambia
887                               Yemen
732                      Western Sahara
876                Wallis and Futuna Islands
850                     Virgin Islands, US
704                            Viet Nam
862          Venezuelaÿ(Bolivarian Republic)
548                           Vanuatu
860                         Uzbekistan
858                          Uruguay
840                United States of America
826                      United Kingdom
784                  United Arab Emirates
804                           Ukraine
800                            Uganda
581                US Minor Outlying Islands
798                            Tuvalu
796                Turks and Caicos Islands
795                       Turkmenistan
792                             Turkey
788                            Tunisia
780                Trinidad and Tobago
776                              Tonga
772                            Tokelau
768                               Togo
626                          Timor-Leste
764                           Thailand
834          Tanzania, United Republic of
762                         Tajikistan
...                                ...
74                        Bouvet Island
72                             Botswana
70                Bosnia and Herzegovina
68                              Bolivia
64                               Bhutan
```

```
64                               Bhutan
60                             Bermuda
204                              Benin
84                               Belize
56                               Belgium
112                              Belarus
52                             Barbados
50                          Bangladesh
48                               Bahrain
```

```
44                               Bahamas
31                            Azerbaijan
40                               Austria
36                             Australia
533                              Aruba
51                             Armenia
32                           Argentina
28                  Antigua and Barbuda
10                           Antarctica
660                            Anguilla
24                              Angola
20                             Andorra
16                      American Samoa
12                             Algeria
8                              Albania
248                        Aland Islands
4                           Afghanistan

[247 rows x 1 columns]
====================================================
CSV to HORUS - Done
```

b) XML to Horus

Code:

```
import pandas as pd
import xml.etree.ElementTree as ET
def df2xml(data):
header = data.columns
root = ET.Element('root')
for row in range(data.shape[0]):
entry = ET.SubElement(root,'entry')
for index in range(data.shape[1]):
schild=str(header[index])
child = ET.SubElement(entry, schild)
if str(data[schild][row]) != 'nan':
child.text = str(data[schild][row])
else:
child.text = 'n/a'
entry.append(child)
result = ET.tostring(root)
return result
def xml2df(xml_data):
root = ET.XML(xml_data)
all_records = []
for i, child in enumerate(root):
record = {}
for subchild in child:
record[subchild.tag] = subchild.text
all_records.append(record)
return pd.DataFrame(all_records)
sInputFileName='C:/VKHCG/05-DS/9999-Data/Country_Code.xml'
InputData = open(sInputFileName).read()
print('=================================================')
print('Input Data Values ===============================')
print('=================================================')
 print(InputData)
print('=================================================')
#=============================================================
# Processing Rules ===========================================
#=============================================================
ProcessDataXML=InputData
# XML to Data Frame
ProcessData=xml2df(ProcessDataXML)
# Remove columns ISO-2-Code and ISO-3-CODE
ProcessData.drop('ISO-2-CODE', axis=1,inplace=True)
ProcessData.drop('ISO-3-Code',  axis=1,inplace=True)
# Rename Country and ISO-M49
```

```
ProcessData.rename(columns={'Country': 'CountryName'}, inplace=True)
ProcessData.rename(columns={'ISO-M49': 'CountryNumber'}, inplace=True)
# Set new Index
ProcessData.set_index('CountryNumber', inplace=True)
# Sort data by CurrencyNumber
ProcessData.sort_values('CountryName', axis=0, ascending=False, inplace=True)
print('=============================================================')
print('Process Data Values ==================================')
print('=============================================================')
print(ProcessData)
print('=============================================================')
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-XML-Country.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('=============================================================')
print('XML to HORUS - Done')
print('=============================================================')
# Utility done ========================================================
```

Output:

```
In [105]: runfile('H:/VKHCG/05-DS/9999-Data/XML2HORUS.py', wdir='H:/VKHCG/05-DS/9999-Data')

=====================================================
Input Data Values ===================================
=====================================================
<root><entry><Country>Afghanistan</Country><Country>Afghanistan</Country><ISO-2-CODE>AF</
ISO-2-CODE><ISO-2-CODE>AF</ISO-2-CODE><ISO-3-Code>AFG</ISO-3-Code><ISO-3-Code>AFG</ISO-3-
Code><ISO-M49>4</ISO-M49><ISO-M49>4</ISO-M49></entry><entry><Country>Aland Islands</
Country><Country>Aland Islands</Country><ISO-2-CODE>AX</ISO-2-CODE><ISO-2-CODE>AX</ISO-2-
CODE><ISO-3-Code>ALA</ISO-3-Code><ISO-3-Code>ALA</ISO-3-Code><ISO-M49>248</ISO-M49><ISO-
M49>248</ISO-M49></entry><entry><Country>Albania</Country><Country>Albania</Country><ISO-2-
CODE>AL</ISO-2-CODE><ISO-2-CODE>AL</ISO-2-CODE><ISO-3-Code>ALB</ISO-3-Code><ISO-3-Code>ALB</
ISO-3-Code><ISO-M49>8</ISO-M49><ISO-M49>8</ISO-M49></entry><entry><Country>Algeria</
Country><Country>Algeria</Country><ISO-2-CODE>DZ</ISO-2-CODE><ISO-2-CODE>DZ</ISO-2-
CODE><ISO-3-Code>DZA</ISO-3-Code><ISO-3-Code>DZA</ISO-3-Code><ISO-M49>12</ISO-M49><ISO-
M49>12</ISO-M49></entry><entry><Country>American Samoa</Country><Country>American Samoa</
Country><ISO-2-CODE>AS</ISO-2-CODE><ISO-2-CODE>AS</ISO-2-CODE><ISO-3-Code>ASM</ISO-3-
Code><ISO-3-Code>ASM</ISO-3-Code><ISO-M49>16</ISO-M49><ISO-M49>16</ISO-M49></
entry><entry><Country>Andorra</Country><Country>Andorra</Country><ISO-2-CODE>AD</ISO-2-
CODE><ISO-2-CODE>AD</ISO-2-CODE><ISO-3-Code>AND</ISO-3-Code><ISO-3-Code>AND</ISO-3-
Code><ISO-M49>20</ISO-M49><ISO-M49>20</ISO-M49></entry><entry><Country>Angola</
Country><Country>Angola</Country><ISO-2-CODE>AO</ISO-2-CODE><ISO-2-CODE>AO</ISO-2-
CODE><ISO-3-Code>AGO</ISO-3-Code><ISO-3-Code>AGO</ISO-3-Code><ISO-M49>24</ISO-M49><ISO-
M49>24</ISO-M49></entry><entry><Country>Anguilla</Country><Country>Anguilla</Country><ISO-2-
CODE>AI</ISO-2-CODE><ISO-2-CODE>AI</ISO-2-CODE><ISO-3-Code>AIA</ISO-3-Code><ISO-3-Code>AIA</
ISO-3-Code><ISO-M49>660</ISO-M49><ISO-M49>660</ISO-M49></entry><entry><Country>Antarctica</
Country><Country>Antarctica</Country><ISO-2-CODE>AQ</ISO-2-CODE><ISO-2-CODE>AQ</ISO-2-
CODE><ISO-3-Code>ATA</ISO-3-Code><ISO-3-Code>ATA</ISO-3-Code><ISO-M49>10</ISO-M49><ISO-
M49>10</ISO-M49></entry><entry><Country>Antigua and Barbuda</Country><Country>Antigua and
Barbuda</Country><ISO-2-CODE>AG</ISO-2-CODE><ISO-2-CODE>AG</ISO-2-CODE><ISO-3-Code>ATG</
ISO-3-Code><ISO-3-Code>ATG</ISO-3-Code><ISO-M49>28</ISO-M49><ISO-M49>28</ISO-M49></
entry><entry><Country>Argentina</Country><Country>Argentina</Country><ISO-2-CODE>AR</ISO-2-
CODE><ISO-2-CODE>AR</ISO-2-CODE><ISO-3-Code>ARG</ISO-3-Code><ISO-3-Code>ARG</ISO-3-
Code><ISO-M49>32</ISO-M49><ISO-M49>32</ISO-M49></entry><entry><Country>Armenia</
Country><Country>Armenia</Country><ISO-2-CODE>AM</ISO-2-CODE><ISO-2-CODE>AM</ISO-2-
CODE><ISO-3-Code>ARM</ISO-3-Code><ISO-3-Code>ARM</ISO-3-Code><ISO-M49>51</ISO-M49><ISO-
M49>51</ISO-M49></entry><entry><Country>Aruba</Country><Country>Aruba</Country><ISO-2-
CODE>AW</ISO-2-CODE><ISO-2-CODE>AW</ISO-2-CODE><ISO-3-Code>ABW</ISO-3-Code><ISO-3-Code>ABW</
ISO-3-Code><ISO-M49>533</ISO-M49><ISO-M49>533</ISO-M49></entry><entry><Country>Australia</
```

```
Country><Country>Australia</Country><ISO-2-CODE>AU</ISO-2-CODE><ISO-2-CODE>AU</ISO-2-
CODE><ISO-3-Code>AUS</ISO-3-Code><ISO-3-Code>AUS</ISO-3-Code><ISO-M49>36</ISO-M49><ISO-
M49>36</ISO-M49></entry><entry><Country>Austria</Country><Country>Austria</Country><ISO-2-
CODE>AT</ISO-2-CODE><ISO-2-CODE>AT</ISO-2-CODE><ISO-3-Code>AUT</ISO-3-Code><ISO-3-Code>AUT</
ISO-3-Code><ISO-M49>40</ISO-M49><ISO-M49>40</ISO-M49></entry><entry><Country>Azerbaijan</
Country><Country>Azerbaijan</Country><ISO-2-CODE>AZ</ISO-2-CODE><ISO-2-CODE>AZ</ISO-2-
CODE><ISO-3-Code>AZE</ISO-3-Code><ISO-3-Code>AZE</ISO-3-Code><ISO-M49>31</ISO-M49><ISO-
M49>31</ISO-M49></entry><entry><Country>Bahamas</Country><Country>Bahamas</Country><ISO-2-
CODE>BS</ISO-2-CODE><ISO-2-CODE>BS</ISO-2-CODE><ISO-3-Code>BHS</ISO-3-Code><ISO-3-Code>BHS</
ISO-3-Code><ISO-M49>44</ISO-M49><ISO-M49>44</ISO-M49></entry><entry><Country>Bahrain</
Country><Country>Bahrain</Country><ISO-2-CODE>BH</ISO-2-CODE><ISO-2-CODE>BH</ISO-2-
CODE><ISO-3-Code>BHR</ISO-3-Code><ISO-3-Code>BHR</ISO-3-Code><ISO-M49>48</ISO-M49><ISO-
M49>48</ISO-M49></entry><entry><Country>Bangladesh</Country><Country>Bangladesh</
Country><ISO-2-CODE>BD</ISO-2-CODE><ISO-2-CODE>BD</ISO-2-CODE><ISO-3-Code>BGD</ISO-3-
Code><ISO-3-Code>BGD</ISO-3-Code><ISO-M49>50</ISO-M49><ISO-M49>50</ISO-M49></
entry><entry><Country>Barbados</Country><Country>Barbados</Country><ISO-2-CODE>BB</ISO-2-
CODE><ISO-2-CODE>BB</ISO-2-CODE><ISO-3-Code>BRB</ISO-3-Code><ISO-3-Code>BRB</ISO-3-

                                                                                    1
```

```
Code><ISO-M49>52</ISO-M49><ISO-M49>52</ISO-M49></entry><entry><Country>Belarus</
Country><Country>Belarus</Country><ISO-2-CODE>BY</ISO-2-CODE><ISO-2-CODE>BY</ISO-2-
CODE><ISO-3-Code>BLR</ISO-3-Code><ISO-3-Code>BLR</ISO-3-Code><ISO-M49>112</ISO-M49><ISO-
M49>112</ISO-M49></entry><entry><Country>Belgium</Country><Country>Belgium</Country><ISO-2-
CODE>BE</ISO-2-CODE><ISO-2-CODE>BE</ISO-2-CODE><ISO-3-Code>BEL</ISO-3-Code><ISO-3-Code>BEL</
ISO-3-Code><ISO-M49>56</ISO-M49><ISO-M49>56</ISO-M49></entry><entry><Country>Belize</
Country><Country>Belize</Country><ISO-2-CODE>BZ</ISO-2-CODE><ISO-2-CODE>BZ</ISO-2-
CODE><ISO-3-Code>BLZ</ISO-3-Code><ISO-3-Code>BLZ</ISO-3-Code><ISO-M49>84</ISO-M49><ISO-
M49>84</ISO-M49></entry><entry><Country>Benin</Country><Country>Benin</Country><ISO-2-
CODE>BJ</ISO-2-CODE><ISO-2-CODE>BJ</ISO-2-CODE><ISO-3-Code>BEN</ISO-3-Code><ISO-3-Code>BEN</
ISO-3-Code><ISO-M49>204</ISO-M49><ISO-M49>204</ISO-M49></entry><entry><Country>Bermuda</
Country><Country>Bermuda</Country><ISO-2-CODE>BM</ISO-2-CODE><ISO-2-CODE>BM</ISO-2-
CODE><ISO-3-Code>BMU</ISO-3-Code><ISO-3-Code>BMU</ISO-3-Code><ISO-M49>60</ISO-M49><ISO-
```

CODE><ISO-3-Code>BMU</ISO-3-Code><ISO-3-Code>BMU</ISO-3-Code><ISO-M49>60</ISO-M49><ISO-M49>60</ISO-M49></entry><entry><Country>Bhutan</Country><Country>Bhutan</Country><ISO-2-CODE>BT</ISO-2-CODE><ISO-2-CODE>BT</ISO-2-CODE><ISO-3-Code>BTN</ISO-3-Code><ISO-3-Code>BTN</ISO-3-Code><ISO-M49>64</ISO-M49><ISO-M49>64</ISO-M49></entry><entry><Country>Bolivia</Country><Country>Bolivia</Country><ISO-2-CODE>BO</ISO-2-CODE><ISO-2-CODE>BO</ISO-2-CODE><ISO-3-Code>BOL</ISO-3-Code><ISO-3-Code>BOL</ISO-3-Code><ISO-M49>68</ISO-M49><ISO-M49>68</ISO-M49></entry><entry><Country>Bosnia and Herzegovina</Country><Country>Bosnia and Herzegovina</Country><ISO-2-CODE>BA</ISO-2-CODE><ISO-2-CODE>BA</ISO-2-CODE><ISO-3-Code>BIH</ISO-3-Code><ISO-3-Code>BIH</ISO-3-Code><ISO-M49>70</ISO-M49><ISO-M49>70</ISO-M49></entry><entry><Country>Botswana</Country><Country>Botswana</Country><ISO-2-CODE>BW</ISO-2-CODE><ISO-2-CODE>BW</ISO-2-CODE><ISO-3-Code>BWA</ISO-3-Code><ISO-3-Code>BWA</ISO-3-Code><ISO-M49>72</ISO-M49><ISO-M49>72</ISO-M49></entry><entry><Country>Bouvet Island</Country><Country>Bouvet Island</Country><ISO-2-CODE>BV</ISO-2-CODE><ISO-2-CODE>BV</ISO-2-CODE><ISO-3-Code>BVT</ISO-3-Code><ISO-3-Code>BVT</ISO-3-Code><ISO-M49>74</ISO-M49><ISO-M49>74</ISO-M49></entry><entry><Country>Brazil</Country><Country>Brazil</Country><ISO-2-CODE>BR</ISO-2-CODE><ISO-2-CODE>BR</ISO-2-CODE><ISO-3-Code>BRA</ISO-3-Code><ISO-3-Code>BRA</ISO-3-Code><ISO-M49>76</ISO-M49><ISO-M49>76</ISO-M49></entry><entry><Country>British Virgin Islands</Country><Country>British Virgin Islands</Country><ISO-2-CODE>VG</ISO-2-CODE><ISO-2-CODE>VG</ISO-2-CODE><ISO-3-Code>VGB</ISO-3-Code><ISO-3-Code>VGB</ISO-3-Code><ISO-M49>92</ISO-M49><ISO-M49>92</ISO-M49></entry><entry><Country>British Indian Ocean Territory</Country><Country>British Indian Ocean Territory</Country><ISO-2-CODE>IO</ISO-2-CODE><ISO-2-CODE>IO</ISO-2-CODE><ISO-3-Code>IOT</ISO-3-Code><ISO-3-Code>IOT</ISO-3-Code><ISO-M49>86</ISO-M49><ISO-M49>86</ISO-M49></entry><entry><Country>Brunei Darussalam</Country><Country>Brunei Darussalam</Country><ISO-2-CODE>BN</ISO-2-CODE><ISO-2-CODE>BN</ISO-2-CODE><ISO-3-Code>BRN</ISO-3-Code><ISO-3-Code>BRN</ISO-3-Code><ISO-M49>96</ISO-M49><ISO-M49>96</ISO-M49></entry><entry><Country>Bulgaria</Country><Country>Bulgaria</Country><ISO-2-CODE>BG</ISO-2-CODE><ISO-2-CODE>BG</ISO-2-CODE><ISO-3-Code>BGR</ISO-3-Code><ISO-3-Code>BGR</ISO-3-Code><ISO-M49>100</ISO-M49><ISO-M49>100</ISO-M49></entry><entry><Country>Burkina Faso</Country><Country>Burkina Faso</Country><ISO-2-CODE>BF</ISO-2-CODE><ISO-2-CODE>BF</ISO-2-CODE><ISO-3-Code>BFA</ISO-3-Code><ISO-3-Code>BFA</ISO-3-Code><ISO-M49>854</ISO-M49><ISO-M49>854</ISO-M49></entry><entry><Country>Burundi</Country><Country>Burundi</Country><ISO-2-CODE>BI</ISO-2-CODE><ISO-2-CODE>BI</ISO-2-CODE><ISO-3-Code>BDI</ISO-3-Code><ISO-3-Code>BDI</ISO-3-Code><ISO-M49>108</ISO-M49><ISO-M49>108</ISO-M49></entry><entry><Country>Cambodia</Country><Country>Cambodia</Country><ISO-2-CODE>KH</ISO-2-CODE><ISO-2-CODE>KH</ISO-2-CODE><ISO-3-Code>KHM</ISO-3-Code><ISO-3-Code>KHM</ISO-3-Code><ISO-M49>116</ISO-M49><ISO-M49>116</ISO-M49></entry><entry><Country>Cameroon</Country><Country>Cameroon</Country><ISO-2-CODE>CM</ISO-2-CODE><ISO-2-CODE>CM</ISO-2-CODE><ISO-3-Code>CMR</ISO-3-Code><ISO-3-Code>CMR</ISO-3-Code><ISO-M49>120</ISO-M49><ISO-M49>120</ISO-M49></entry><entry><Country>Canada</Country><Country>Canada</Country><ISO-2-CODE>CA</ISO-2-CODE><ISO-2-CODE>CA</ISO-2-CODE><ISO-3-Code>CAN</ISO-3-Code><ISO-3-Code>CAN</ISO-3-Code><ISO-M49>124</ISO-M49><ISO-M49>124</ISO-M49></entry><entry><Country>Cape Verde</Country><Country>Cape Verde</Country><ISO-2-CODE>CV</ISO-2-CODE><ISO-2-CODE>CV</ISO-2-CODE><ISO-3-Code>CPV</ISO-3-Code><ISO-3-Code>CPV</ISO-3-Code><ISO-M49>132</ISO-M49><ISO-M49>132</ISO-M49></entry><entry><Country>Cayman Islands</Country><Country>Cayman Islands</Country><ISO-2-CODE>KY</ISO-2-CODE><ISO-2-CODE>KY</ISO-2-CODE><ISO-3-Code>CYM</ISO-3-Code><ISO-3-Code>CYM</ISO-3-Code><ISO-M49>136</ISO-M49><ISO-M49>136</ISO-M49></entry><entry><Country>Central African Republic</Country><Country>Central African Republic</Country><ISO-2-CODE>CF</ISO-2-CODE><ISO-2-CODE>CF</ISO-2-CODE><ISO-3-Code>CAF</ISO-3-Code><ISO-3-Code>CAF</ISO-3-Code><ISO-M49>140</ISO-M49><ISO-M49>140</ISO-M49></entry><entry><Country>Chad</Country><Country>Chad</Country><ISO-2-CODE>TD</ISO-2-CODE><ISO-2-CODE>TD</ISO-2-CODE><ISO-3-Code>TCD</ISO-3-Code><ISO-3-Code>TCD</ISO-3-Code><ISO-M49>148</ISO-M49><ISO-M49>148</ISO-M49></entry><entry><Country>Chile</Country><Country>Chile</Country><ISO-2-CODE>CL</ISO-2-CODE><ISO-2-CODE>CL</ISO-2-CODE><ISO-3-Code>CHL</ISO-3-Code><ISO-3-Code>CHL</ISO-3-Code><ISO-M49>152</ISO-M49><ISO-M49>152</ISO-M49></entry><entry><Country>China</Country><Country>China</Country><ISO-2-CODE>CN</ISO-2-CODE><ISO-2-CODE>CN</ISO-2-CODE><ISO-3-Code>CHN</ISO-3-Code><ISO-3-Code>CHN</ISO-3-Code><ISO-M49>156</ISO-M49><ISO-M49>156</ISO-M49></entry><entry><Country>Hong Kong, SAR China</Country><Country>Hong Kong, SAR China</Country><ISO-2-CODE>HK</ISO-2-CODE><ISO-2-CODE>HK</ISO-2-CODE><ISO-3-Code>HKG</ISO-3-Code><ISO-3-Code>HKG</ISO-3-Code><ISO-M49>344</ISO-M49><ISO-M49>344</ISO-M49></entry><entry><Country>Macao, SAR China</Country><Country>Macao, SAR China</Country><ISO-2-CODE>MO</ISO-2-CODE><ISO-2-CODE>MO</ISO-2-CODE><ISO-3-Code>MAC</ISO-3-Code><ISO-3-Code>MAC</ISO-3-Code><ISO-M49>446</ISO-M49><ISO-M49>446</ISO-M49></entry><entry><Country>Christmas Island</Country><Country>Christmas Island</Country><ISO-2-CODE>CX</ISO-2-CODE><ISO-2-CODE>CX</ISO-2-CODE><ISO-3-Code>CXR</ISO-3-Code><ISO-3-Code>CXR</ISO-3-Code><ISO-M49>162</ISO-M49><ISO-M49>166</ISO-M49></entry><entry><Country>Colombia</Country><Country>Colombia</Country><ISO-2-CODE>CO</ISO-2-CODE><ISO-2-CODE>CO</ISO-2-CODE><ISO-3-Code>COL</ISO-3-Code><ISO-3-Code>COL</ISO-3-Code><ISO-M49>170</ISO-M49><ISO-M49>170</ISO-M49></entry><entry><Country>Comoros</Country><Country>Comoros</Country><ISO-2-CODE>KM</ISO-2-CODE><ISO-2-CODE>KM</ISO-2-CODE><ISO-3-Code>COM</ISO-3-Code><ISO-3-Code>COM</ISO-3-Code><ISO-M49>174</ISO-M49><ISO-M49>174</ISO-M49></entry><entry><Country>Congoÿ(Brazzaville)</Country><Country>Congoÿ(Brazzaville)</Country><ISO-2-CODE>CG</ISO-2-CODE><ISO-2-CODE>CG</ISO-2-CODE><ISO-3-Code>COG</ISO-3-Code><ISO-3-Code>COG</ISO-3-Code><ISO-M49>178</ISO-M49><ISO-M49>178</ISO-M49></entry><entry><Country>Congo, (Kinshasa)</Country><Country>Congo, (Kinshasa)</Country><ISO-2-CODE>CD</ISO-2-CODE><ISO-2-CODE>CD</ISO-2-CODE><ISO-3-Code>COD</ISO-3-Code><ISO-3-Code>COD</ISO-3-Code><ISO-M49>180</ISO-M49><ISO-M49>180</ISO-M49></entry><entry><Country>Cook Islands</Country><Country>Cook Islands</Country><ISO-2-CODE>CK</ISO-2-CODE><ISO-2-CODE>CK</ISO-2-CODE><ISO-3-Code>COK</ISO-3-Code><ISO-3-Code>COK</ISO-3-Code><ISO-M49>184</ISO-M49><ISO-M49>184</ISO-M49></entry><entry><Country>Costa Rica</Country><Country>Costa Rica</Country><ISO-2-CODE>CR</ISO-2-CODE><ISO-2-CODE>CR</ISO-2-CODE><ISO-3-Code>CRI</ISO-3-Code><ISO-3-Code>CRI</ISO-3-Code><ISO-M49>188</ISO-M49><ISO-M49>188</ISO-M49></entry><entry><Country>C“te d'Ivoire</Country><Country>C“te d'Ivoire</Country><ISO-2-CODE>CI</ISO-2-CODE><ISO-2-CODE>CI</ISO-2-CODE><ISO-3-Code>CIV</ISO-3-Code><ISO-3-Code>CIV</ISO-3-Code><ISO-M49>384</ISO-M49><ISO-M49>384</ISO-M49></entry><entry><Country>Croatia</Country><Country>Croatia</Country><ISO-2-CODE>HR</ISO-2-CODE><ISO-2-CODE>HR</ISO-2-CODE><ISO-3-Code>HRV</ISO-3-Code><ISO-3-Code>HRV</ISO-3-Code><ISO-M49>191</ISO-M49><ISO-M49>191</ISO-M49></entry><entry><Country>Cuba</Country><Country>Cuba</Country><ISO-2-CODE>CU</ISO-2-CODE><ISO-2-CODE>CU</ISO-2-CODE><ISO-3-Code>CUB</ISO-3-Code><ISO-3-Code>CUB</ISO-3-Code><ISO-M49>192</ISO-M49><ISO-M49>192</ISO-M49></entry><entry><Country>Cyprus</Country><Country>Cyprus</Country><ISO-2-CODE>CY</ISO-2-CODE><ISO-2-CODE>CY</ISO-2-CODE><ISO-3-Code>CYP</ISO-3-Code><ISO-3-Code>CYP</ISO-3-Code><ISO-M49>196</ISO-M49><ISO-M49>196</ISO-M49></entry><entry><Country>Czech Republic</Country><Country>Czech Republic</Country><ISO-2-CODE>CZ</ISO-2-CODE><ISO-2-CODE>CZ</ISO-2-CODE><ISO-3-Code>CZE</ISO-3-Code><ISO-3-Code>CZE</ISO-3-Code><ISO-M49>203</ISO-M49><ISO-M49>203</ISO-M49></entry><entry><Country>Denmark</Country><Country>Denmark</Country><ISO-2-CODE>DK</ISO-2-CODE><ISO-2-CODE>DK</ISO-2-CODE><ISO-3-Code>DNK</ISO-3-Code><ISO-3-Code>DNK</ISO-3-Code><ISO-M49>208</ISO-M49><ISO-M49>208</ISO-M49></entry><entry><Country>Djibouti</Country><Country>Djibouti</Country><ISO-2-CODE>DJ</ISO-2-CODE><ISO-2-CODE>DJ</ISO-2-CODE><ISO-3-Code>DJI</ISO-3-Code><ISO-3-Code>DJI</ISO-3-Code><ISO-M49>262</ISO-M49><ISO-M49>262</ISO-M49></entry><entry><Country>Dominica</Country><Country>Dominica</Country><ISO-2-CODE>DM</ISO-2-CODE><ISO-2-CODE>DM</ISO-2-CODE><ISO-3-Code>DMA</ISO-3-Code><ISO-3-Code>DMA</ISO-3-Code><ISO-M49>212</ISO-M49><ISO-M49>212</ISO-M49></entry><entry><Country>Dominican Republic</Country><Country>Dominican Republic</Country><ISO-2-CODE>DO</ISO-2-CODE><ISO-2-CODE>DO</ISO-2-CODE><ISO-3-Code>DOM</ISO-3-Code><ISO-3-Code>DOM</ISO-3-Code><ISO-M49>214</ISO-M49><ISO-M49>214</ISO-M49></entry><entry><Country>Ecuador</Country><Country>Ecuador</Country><ISO-2-CODE>EC</ISO-2-CODE><ISO-2-CODE>EC</ISO-2-CODE><ISO-3-Code>ECU</ISO-3-Code><ISO-3-Code>ECU</ISO-3-Code><ISO-M49>218</ISO-M49><ISO-M49>218</ISO-M49></entry><entry><Country>Egypt</Country><Country>Egypt</Country><ISO-2-CODE>EG</ISO-2-CODE><ISO-2-CODE>EG</ISO-2-CODE><ISO-3-Code>EGY</ISO-3-Code><ISO-3-Code>EGY</ISO-3-Code><ISO-M49>818</ISO-M49><ISO-M49>818</ISO-M49></entry><entry><Country>El Salvador</Country><Country>El Salvador</Country><ISO-2-CODE>SV</ISO-2-CODE><ISO-2-CODE>SV</ISO-2-CODE><ISO-3-Code>SLV</ISO-3-Code><ISO-3-Code>SLV</ISO-3-Code><ISO-M49>222</ISO-M49><ISO-M49>222</ISO-M49></entry><entry><Country>Equatorial Guinea</Country><Country>Equatorial Guinea</Country><ISO-2-CODE>GQ</ISO-2-CODE><ISO-2-CODE>GQ</ISO-2-CODE><ISO-3-Code>GNQ</ISO-3-Code><ISO-3-Code>GNQ</ISO-3-Code><ISO-M49>226</ISO-M49><ISO-M49>226</ISO-M49></entry><entry><Country>Eritrea</Country><Country>Eritrea</Country><ISO-2-CODE>ER</ISO-2-CODE><ISO-2-CODE>ER</ISO-2-CODE><ISO-3-Code>ERI</ISO-3-Code><ISO-3-Code>ERI</ISO-3-Code><ISO-M49>232</ISO-M49><ISO-M49>232</ISO-M49></entry><entry><Country>Estonia</Country><Country>Estonia</Country><ISO-2-CODE>EE</ISO-2-CODE><ISO-2-CODE>EE</ISO-2-CODE><ISO-3-Code>EST</ISO-3-Code><ISO-3-Code>EST</ISO-3-Code><ISO-M49>233</ISO-M49><ISO-M49>233</ISO-M49></entry><entry><Country>Ethiopia</Country><Country>Ethiopia</Country><ISO-2-CODE>ET</ISO-2-CODE><ISO-2-CODE>ET</ISO-2-CODE><ISO-3-Code>ETH</ISO-3-Code><ISO-3-Code>ETH</ISO-3-Code><ISO-M49>231</ISO-M49><ISO-M49>231</ISO-M49></entry><entry><Country>Falkland Islands (Malvinas)</Country><Country>Falkland Islands (Malvinas)</Country><ISO-2-CODE>FK</ISO-2-CODE><ISO-2-CODE>FK</ISO-2-CODE><ISO-3-Code>FLK</ISO-3-Code><ISO-3-Code>FLK</ISO-3-Code><ISO-M49>238</ISO-M49><ISO-M49>238</ISO-M49></entry><entry><Country>Faroe Islands</Country><Country>Faroe Islands</Country><ISO-2-CODE>FO</ISO-2-CODE><ISO-2-CODE>FO</ISO-2-CODE><ISO-3-Code>FRO</ISO-3-Code><ISO-3-Code>FRO</ISO-3-Code><ISO-M49>234</ISO-M49><ISO-M49>234</ISO-M49></entry><entry><Country>Fiji</Country><Country>Fiji</Country><ISO-2-CODE>FJ</ISO-2-CODE><ISO-2-CODE>FJ</ISO-2-CODE><ISO-3-Code>FJI</ISO-3-Code><ISO-3-Code>FJI</ISO-3-Code><ISO-M49>242</ISO-M49><ISO-M49>242</ISO-M49></entry><entry><Country>Finland</Country><Country>Finland</Country><ISO-2-CODE>FI</ISO-2-CODE><ISO-2-CODE>FI</ISO-2-CODE><ISO-3-Code>FIN</ISO-3-Code><ISO-3-Code>FIN</ISO-3-Code><ISO-M49>246</ISO-M49><ISO-M49>246</ISO-M49></entry><entry><Country>France</Country><Country>France</Country><ISO-2-CODE>FR</ISO-2-CODE><ISO-2-CODE>FR</ISO-2-CODE><ISO-3-Code>FRA</ISO-3-Code><ISO-3-Code>FRA</ISO-3-Code><ISO-M49>250</ISO-M49><ISO-M49>250</ISO-M49></entry><entry><Country>French Guiana</Country><Country>French Guiana</Country><ISO-2-CODE>GF</ISO-2-CODE><ISO-2-CODE>GF</ISO-2-CODE><ISO-3-Code>GUF</ISO-3-Code><ISO-3-Code>GUF</ISO-3-Code><ISO-M49>254</ISO-M49><ISO-M49>254</ISO-M49></entry><entry><Country>French Polynesia</Country><Country>French Polynesia</Country><ISO-2-CODE>PF</ISO-2-CODE><ISO-2-CODE>PF</ISO-2-CODE><ISO-3-Code>PYF</ISO-3-Code><ISO-3-Code>PYF</ISO-3-Code><ISO-M49>258</ISO-M49><ISO-M49>258</ISO-M49></entry><entry><Country>French Southern Territories</Country><Country>French Southern Territories</Country><ISO-2-CODE>TF</ISO-2-CODE><ISO-2-CODE>TF</ISO-2-CODE><ISO-3-Code>ATF

c) JSON to HORUS Format

Code:

```
# Utility Start JSON to HORUS ================================
# Standard Tools
#=============================================================
import pandas as pd
# Input Agreement ============================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/Country_Code.json'
InputData=pd.read_json(sInputFileName, orient='index', encoding="latin-1")
print('Input Data Values ===================================')
print(InputData)
print('=====================================================')
# Processing Rules ===========================================
ProcessData=InputData
# Remove columns ISO-2-Code and ISO-3-CODE
ProcessData.drop('ISO-2-CODE', axis=1,inplace=True)
ProcessData.drop('ISO-3-Code',  axis=1,inplace=True)
# Rename Country and ISO-M49
ProcessData.rename(columns={'Country': 'CountryName'}, inplace=True)
ProcessData.rename(columns={'ISO-M49': 'CountryNumber'}, inplace=True)
# Set new Index
ProcessData.set_index('CountryNumber', inplace=True)
# Sort data by CurrencyNumber
ProcessData.sort_values('CountryName', axis=0, ascending=False, inplace=True)
print('Process Data Values ================================')
print(ProcessData)
print('=====================================================')
# Output Agreement ===========================================
OutputData=ProcessData
sOutputFileName='c:/VKHCG/05-DS/9999-Data/HORUS-JSON-Country.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('JSON to HORUS - Done')
# Utility done ===============================================
```

Output:

```
In [117]: runfile('H:/VKHCG/05-DS/9999-Data/JSON2HORUS.py', wdir='H:/VKHCG/05-DS/9999-Data')

Input Data Values ===================================
                        Country ISO-2-CODE ISO-3-Code  ISO-M49
0                   Afghanistan         AF         AFG        4
1                 Aland Islands         AX         ALA      248
2                       Albania         AL         ALB        8
3                       Algeria         DZ         DZA       12
4                American Samoa         AS         ASM       16
5                       Andorra         AD         AND       20
6                        Angola         AO         AGO       24
7                      Anguilla         AI         AIA      660
8                    Antarctica         AQ         ATA       10
9             Antigua and Barbuda         AG         ATG       28
10                    Argentina         AR         ARG       32
11                      Armenia         AM         ARM       51
12                        Aruba         AW         ABW      533
13                    Australia         AU         AUS       36
14                      Austria         AT         AUT       40
15                   Azerbaijan         AZ         AZE       31
16                      Bahamas         BS         BHS       44
17                      Bahrain         BH         BHR       48
18                   Bangladesh         BD         BGD       50
19                     Barbados         BB         BRB       52
20                      Belarus         BY         BLR      112
21                      Belgium         BE         BEL       56
22                       Belize         BZ         BLZ       84
23                        Benin         BJ         BEN      204
24                      Bermuda         BM         BMU       60
25                       Bhutan         BT         BTN       64
26                      Bolivia         BO         BOL       68
27      Bosnia and Herzegovina          BA         BIH       70
28                     Botswana         BW         BWA       72
29                Bouvet Island         BV         BVT       74
..                          ...        ...         ...      ...
217                  Tajikistan         TJ         TJK      762
218  Tanzania, United Republic of         TZ         TZA      834
219                    Thailand         TH         THA      764
220                  Timor-Leste         TL         TLS      626
221                        Togo         TG         TGO      768
222                     Tokelau         TK         TKL      772
223                       Tonga         TO         TON      776
224         Trinidad and Tobago         TT         TTO      780
225                     Tunisia         TN         TUN      788
226                      Turkey         TR         TUR      792
227                Turkmenistan         TM         TKM      795
228  Turks and Caicos Islands          TC         TCA      796
229                      Tuvalu         TV         TUV      798
230                      Uganda         UG         UGA      800
231                     Ukraine         UA         UKR      804
232          United Arab Emirates        AE         ARE      784
233              United Kingdom          GB         GBR      826
234   United States of America           US         USA      840
235  US Minor Outlying Islands           UM         UMI      581
236                     Uruguay         UY         URY      858
237                  Uzbekistan         UZ         UZB      860
238                     Vanuatu         VU         VUT      548
239  Venezuelaÿ(Bolivarian Republic)    VE         VEN      862
240                   Viet Nam          VN         VNM      704
241          Virgin Islands, US         VI         VIR      850
242  Wallis and Futuna Islands          WF         WLF      876
243             Western Sahara          EH         ESH      732
244                      Yemen          YE         YEM      887
245                     Zambia          ZM         ZMB      894
246                   Zimbabwe          ZW         ZWE      716

[247 rows x 4 columns]
=====================================================

Process Data Values ================================
                                CountryName
CountryNumber
716                                Zimbabwe
894                                 Zambia
887                                  Yemen
732                         Western Sahara
876                 Wallis and Futuna Islands
850                       Virgin Islands, US
704                               Viet Nam
862           Venezuelaÿ(Bolivarian Republic)
548                               Vanuatu
860                             Uzbekistan
858                               Uruguay
840                 United States of America
826                        United Kingdom
784                    United Arab Emirates
804                               Ukraine
800                                Uganda
581                 US Minor Outlying Islands
798                               Tuvalu
796                 Turks and Caicos Islands
795                            Turkmenistan
792                               Turkey
788                               Tunisia
780                     Trinidad and Tobago
776                               Tonga
772                             Tokelau
768                                Togo
626                           Timor-Leste
764                             Thailand
834           Tanzania, United Republic of
762                           Tajikistan
...                                  ...
74                        Bouvet Island
72                              Botswana
70                Bosnia and Herzegovina
68                              Bolivia
64                               Bhutan
60                              Bermuda
204                               Benin
84                               Belize
64                               Bhutan
60                              Bermuda
204                               Benin
84                               Belize
56                              Belgium
112                             Belarus
52                             Barbados
50                           Bangladesh
48                              Bahrain
44                              Bahamas
31                           Azerbaijan
40                              Austria
36                            Australia
533                               Aruba
51                              Armenia
32                            Argentina
28                  Antigua and Barbuda
10                           Antarctica
660                            Anguilla
24                              Angola
20                             Andorra
16                      American Samoa
12                             Algeria
8                              Albania
248                        Aland Islands
4                           Afghanistan

[247 rows x 1 columns]
=====================================================
JSON to HORUS - Done
```

d) MySql Database to HORUS Format

Code :

```
import pandas as pd
import sqlite3 as sq
# Input Agreement ============================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/utility.db'
sInputTable='Country_Code'
conn = sq.connect(sInputFileName)
sSQL='select * FROM ' + sInputTable + ';'
InputData=pd.read_sql_query(sSQL,  conn)
print('Input Data Values ===================================')
print(InputData)
print('=====================================================')
# Processing Rules ===========================================
ProcessData=InputData
# Remove columns ISO-2-Code and ISO-3-CODE
ProcessData.drop('ISO-2-CODE', axis=1,inplace=True)
ProcessData.drop('ISO-3-Code',  axis=1,inplace=True)
# Rename Country and ISO-M49
ProcessData.rename(columns={'Country': 'CountryName'}, inplace=True)
ProcessData.rename(columns={'ISO-M49': 'CountryNumber'}, inplace=True)
# Set new Index
ProcessData.set_index('CountryNumber', inplace=True)
# Sort data by CurrencyNumber
ProcessData.sort_values('CountryName', axis=0, ascending=False, inplace=True)
print('Process Data Values ================================')
print(ProcessData)
print('=====================================================')
# Output Agreement ===========================================
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-CSV-Country.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('Database to HORUS - Done')
# Utility done ===============================================
```

Output:

```
In [6]: runfile('C:/VKHCG/05-DS/9999-Data/DATABASE2HORUS.py', wdir='C:/VKHCG/05-DS/9999-Data')


Input Data Values ==================================
     index                         Country ISO-2-CODE ISO-3-Code  ISO-M49
0        0                     Afghanistan         AF        AFG        4
1        1                   Aland Islands         AX        ALA      248
2        2                         Albania         AL        ALB        8
3        3                         Algeria         DZ        DZA       12
4        4                  American Samoa         AS        ASM       16
5        5                         Andorra         AD        AND       20
6        6                          Angola         AO        AGO       24
7        7                        Anguilla         AI        AIA      660
8        8                      Antarctica         AQ        ATA       10
9        9              Antigua and Barbuda         AG        ATG       28
10      10                       Argentina         AR        ARG       32
11      11                         Armenia         AM        ARM       51
12      12                           Aruba         AW        ABW      533
13      13                       Australia         AU        AUS       36
14      14                         Austria         AT        AUT       40
15      15                      Azerbaijan         AZ        AZE       31
16      16                         Bahamas         BS        BHS       44
17      17                         Bahrain         BH        BHR       48
18      18                      Bangladesh         BD        BGD       50
19      19                        Barbados         BB        BRB       52
20      20                         Belarus         BY        BLR      112
21      21                         Belgium         BE        BEL       56
22      22                          Belize         BZ        BLZ       84
23      23                           Benin         BJ        BEN      204
24      24                         Bermuda         BM        BMU       60
25      25                          Bhutan         BT        BTN       64
26      26                         Bolivia         BO        BOL       68
27      27          Bosnia and Herzegovina         BA        BIH       70
28      28                        Botswana         BW        BWA       72

29      29                  Bouvet Island         BV        BVT       74
..     ...                            ...        ...        ...      ...
217    217                     Tajikistan         TJ        TJK      762
218    218  Tanzania, United Republic of         TZ        TZA      834
219    219                       Thailand         TH        THA      764
220    220                    Timor-Leste         TL        TLS      626
221    221                           Togo         TG        TGO      768
222    222                        Tokelau         TK        TKL      772
223    223                          Tonga         TO        TON      776
224    224            Trinidad and Tobago         TT        TTO      780
225    225                        Tunisia         TN        TUN      788
226    226                         Turkey         TR        TUR      792
227    227                   Turkmenistan         TM        TKM      795
228    228      Turks and Caicos Islands         TC        TCA      796
229    229                         Tuvalu         TV        TUV      798
230    230                         Uganda         UG        UGA      800
231    231                        Ukraine         UA        UKR      804
232    232            United Arab Emirates         AE        ARE      784
233    233                 United Kingdom         GB        GBR      826
234    234      United States of America         US        USA      840
235    235     US Minor Outlying Islands         UM        UMI      581
236    236                        Uruguay         UY        URY      858
237    237                     Uzbekistan         UZ        UZB      860
238    238                        Vanuatu         VU        VUT      548
239    239  Venezuelaÿ(Bolivarian Republic)         VE        VEN      862
240    240                       Viet Nam         VN        VNM      704
241    241             Virgin Islands, US         VI        VIR      850
242    242     Wallis and Futuna Islands         WF        WLF      876
243    243                Western Sahara         EH        ESH      732
244    244                         Yemen         YE        YEM      887
245    245                        Zambia         ZM        ZMB      894
246    246                      Zimbabwe         ZW        ZWE      716

[247 rows x 5 columns]
====================================================
```

```
Process Data Values ================================
              index                      CountryName
CountryNumber
716             246                         Zimbabwe
894             245                           Zambia
887             244                            Yemen
732             243                   Western Sahara
876             242       Wallis and Futuna Islands
850             241             Virgin Islands, US
704             240                         Viet Nam
862             239  Venezuelaÿ(Bolivarian Republic)
548             238                          Vanuatu
860             237                       Uzbekistan
858             236                          Uruguay
840             234        United States of America
826             233                  United Kingdom
784             232            United Arab Emirates
804             231                         Ukraine
800             230                          Uganda
581             235       US Minor Outlying Islands
798             229                          Tuvalu
796             228        Turks and Caicos Islands
795             227                    Turkmenistan
792             226                          Turkey
788             225                         Tunisia
780             224             Trinidad and Tobago
776             223                           Tonga
772             222                         Tokelau
768             221                            Togo
626             220                      Timor-Leste
764             219                        Thailand
834             218     Tanzania, United Republic of
762             217                      Tajikistan
...             ...                             ...
74               29                  Bouvet Island
72               28                       Botswana
70               27          Bosnia and Herzegovina
70              27          Bosnia and Herzegovina
68              26                         Bolivia
64              25                          Bhutan
60              24                         Bermuda
204             23                           Benin
84              22                          Belize
56              21                         Belgium
112             20                         Belarus
52              19                        Barbados
50              18                      Bangladesh
48              17                         Bahrain
44              16                         Bahamas
31              15                      Azerbaijan
40              14                         Austria
36              13                       Australia
533             12                           Aruba
51              11                         Armenia
32              10                       Argentina
28               9             Antigua and Barbuda
10               8                      Antarctica
660              7                        Anguilla
24               6                          Angola
20               5                         Andorra
16               4                  American Samoa
12               3                         Algeria
8                2                         Albania
248              1                     Aland Islands
4                0                     Afghanistan

[247 rows x 2 columns]
====================================================
Database to HORUS - Done
```

e) Picture (JPEG) to HORUS Format

Code:

```
# Utility Start Picture to HORUS ================================
# Standard Tools
#================================================================
from scipy.misc import imread
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
# Input Agreement ===============================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/Angus.jpg'
InputData = imread(sInputFileName, flatten=False, mode='RGBA')
print('Input Data Values ===================================')
print('X: ',InputData.shape[0])
print('Y: ',InputData.shape[1])
print('RGBA: ', InputData.shape[2])
print('=====================================================')
# Processing Rules ==============================================
ProcessRawData=InputData.flatten()
y=InputData.shape[2] + 2
x=int(ProcessRawData.shape[0]/y)
ProcessData=pd.DataFrame(np.reshape(ProcessRawData, (x, y)))
sColumns= ['XAxis','YAxis','Red', 'Green', 'Blue','Alpha']
ProcessData.columns=sColumns
ProcessData.index.names =['ID']
print('Rows: ',ProcessData.shape[0])
print('Columns :',ProcessData.shape[1])
print('=====================================================')
print('Process Data Values ================================')
print('=====================================================')
plt.imshow(InputData)
plt.show()
print('=====================================================')
# Output Agreement ==============================================
OutputData=ProcessData
print('Storing File')
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-Picture.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('=====================================================')
print('Picture to HORUS - Done')
print('=====================================================')
```

f) i. Video to HORUS Format

Code:

Movies to Frames

```
import os
import shutil
import cv2
#============================================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/dog.mp4'
sDataBaseDir='C:/VKHCG/05-DS/9999-Data/temp'
if os.path.exists(sDataBaseDir):
shutil.rmtree(sDataBaseDir)
if not os.path.exists(sDataBaseDir):
os.makedirs(sDataBaseDir)
print('============================================================')
print('Start Movie to Frames')
print('============================================================')
vidcap = cv2.VideoCapture(sInputFileName)
success,image = vidcap.read()
count = 0
while success:
success,image = vidcap.read()
sFrame=sDataBaseDir + str('/dog-frame-' + str(format(count, '04d'))+ '.jpg')
print('Extracted: ', sFrame)
cv2.imwrite(sFrame, image)
if os.path.getsize(sFrame) == 0:
count += -1
os.remove(sFrame)
print('Removed: ', sFrame)
if cv2.waitKey(10) == 27: # exit if Escape is hit
break
count += 1
print('============================================================')
print('Generated : ', count, ' Frames')
print('============================================================')
print('Movie to Frames HORUS - Done')
print('============================================================')
# Utility done ==============================================
```

Output:

```
In [72]: runfile('C:/VKHCG/05-DS/9999-Data/MOVIE2HORUSFrame.py', wdir='C:/VKHCG/
05-DS/9999-Data')


====================================================
Start Movie to Frames
====================================================
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0000.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0001.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0002.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0003.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0004.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0005.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0006.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0007.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0008.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0009.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0010.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0011.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0012.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0013.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0014.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0015.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0016.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0017.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0018.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0019.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0020.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0021.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0022.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0023.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0024.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0025.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0026.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0027.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0028.jpg

Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0079.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0080.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0081.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0082.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0083.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0084.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0085.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0086.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0087.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0088.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0089.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0090.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0091.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0092.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0093.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0094.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0095.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0096.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0097.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0098.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0099.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0100.jpg
Extracted:  C:/VKHCG/05-DS/9999-Data/temp/dog-frame-0101.jpg
====================================================
Generated :  101  Frames
====================================================
Movie to Frames HORUS - Done
====================================================
```

ii. Frames to Horus

```
from scipy.misc import imread
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os
# Input Agreement ============================================
sDataBaseDir='C:/VKHCG/05-DS/9999-Data/temp'
f=0
for file in os.listdir(sDataBaseDir):
if file.endswith(".jpg"):
f += 1
sInputFileName=os.path.join(sDataBaseDir, file)
print('Process : ', sInputFileName)
InputData = imread(sInputFileName, flatten=False, mode='RGBA')
print('Input Data Values ===================================')
print('X: ',InputData.shape[0])
print('Y: ',InputData.shape[1])
print('RGBA: ', InputData.shape[2])
print('=====================================================')
# Processing Rules ===========================================
ProcessRawData=InputData.flatten()
y=InputData.shape[2] + 2
x=int(ProcessRawData.shape[0]/y)
ProcessFrameData=pd.DataFrame(np.reshape(ProcessRawData, (x, y)))
ProcessFrameData['Frame']=file
print('=====================================================')
print('Process Data Values ================================')
print('=====================================================')
plt.imshow(InputData)
plt.show()
if f == 1:
ProcessData=ProcessFrameData
else:
ProcessData=ProcessData.append(ProcessFrameData)
if f > 0:
sColumns= ['XAxis','YAxis','Red', 'Green', 'Blue','Alpha','FrameName']
ProcessData.columns=sColumns
print('=====================================================')
ProcessFrameData.index.names =['ID']
print('Rows: ',ProcessData.shape[0])
print('Columns :',ProcessData.shape[1])
print('=====================================================')
# Output Agreement ===========================================
```

g) Audio to HORUS Format

Code:

```
# Utility Start Audio to HORUS ================================
# Standard Tools
#=============================================================
from scipy.io import wavfile
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
#=============================================================
def show_info(aname, a,r):
print ('.....')
print ("Audio:", aname)
print ('.....')
print ("Rate:", r)
print ('.....')
print ("shape:", a.shape)
print ("dtype:", a.dtype)
print ("min, max:", a.min(), a.max())
print ('.....')
plot_info(aname, a,r)
#=============================================================
def plot_info(aname, a,r):
sTitle= 'Signal Wave - '+ aname + ' at ' + str(r) + 'hz'
plt.title(sTitle)
sLegend=[]
for c in range(a.shape[1]):
sLabel = 'Ch' + str(c+1)
sLegend=sLegend+[str(c+1)]
plt.plot(a[:,c], label=sLabel)
plt.legend(sLegend)
plt.show()
#=============================================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/2ch-sound.wav'
print('=======================================================')
print('Processing : ', sInputFileName)
print('=======================================================')
InputRate, InputData = wavfile.read(sInputFileName)
show_info("2 channel", InputData,InputRate)
ProcessData=pd.DataFrame(InputData)
sColumns= ['Ch1','Ch2']
ProcessData.columns=sColumns
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-Audio-2ch.csv'
OutputData.to_csv(sOutputFileName, index = False)
```

```
#============================================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/4ch-sound.wav'
print('============================================================')
print('Processing : ', sInputFileName)
print('============================================================')
InputRate, InputData = wavfile.read(sInputFileName)
show_info("4 channel", InputData,InputRate)
ProcessData=pd.DataFrame(InputData)
sColumns= ['Ch1','Ch2','Ch3', 'Ch4']
ProcessData.columns=sColumns
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-Audio-4ch.csv'
OutputData.to_csv(sOutputFileName, index = False)
#============================================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/6ch-sound.wav'
print('============================================================')
print('Processing : ', sInputFileName)
print('============================================================')
InputRate, InputData = wavfile.read(sInputFileName)
show_info("6 channel", InputData,InputRate)
ProcessData=pd.DataFrame(InputData)
sColumns= ['Ch1','Ch2','Ch3', 'Ch4', 'Ch5','Ch6']
ProcessData.columns=sColumns
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-Audio-6ch.csv'
OutputData.to_csv(sOutputFileName, index = False)
#============================================================
sInputFileName='C:/VKHCG/05-DS/9999-Data/8ch-sound.wav'
print('============================================================')
print('Processing : ', sInputFileName)
print('============================================================')
InputRate, InputData = wavfile.read(sInputFileName)
show_info("8 channel", InputData,InputRate)
ProcessData=pd.DataFrame(InputData)
sColumns= ['Ch1','Ch2','Ch3', 'Ch4', 'Ch5','Ch6','Ch7','Ch8']
ProcessData.columns=sColumns
OutputData=ProcessData
sOutputFileName='C:/VKHCG/05-DS/9999-Data/HORUS-Audio-8ch.csv'
OutputData.to_csv(sOutputFileName, index = False)
print('============================================================')
print('Audio to HORUS - Done')
```

Practical 3

Utilities and Auditing

A. Fixers Utilities:

Fixers enable your solution to take your existing data and fix a specific quality issue.

```
#---------------------------- Program to Demonstrate Fixers utilities ------------------
import string
import datetime as dt
# 1 Removing leading or lagging spaces from a data entry
print('#1 Removing leading or lagging spaces from a data entry');
baddata = " Data Science with too many spaces is bad!!! "
print('>',baddata,'<')
cleandata=baddata.strip()
print('>',cleandata,'<')


# 2 Removing nonprintable characters from a data entry
print('#2 Removing nonprintable characters from a data entry')
printable = set(string.printable)
baddata = "Data\x00Science with\x02 funny characters is \x10bad!!!"
cleandata=''.join(filter(lambda x: x in string.printable,baddata))
print('Bad Data : ',baddata);
print('Clean Data : ',cleandata)


# 3 Reformatting data entry to match specific formatting criteria.
# Convert YYYY/MM/DD to DD Month YYYY
print('# 3 Reformatting data entry to match specific formatting criteria.')
baddate = dt.date(2019, 10, 31)
baddata=format(baddate,'%Y-%m-%d')
gooddate = dt.datetime.strptime(baddata,'%Y-%m-%d')
gooddata=format(gooddate,'%d %B %Y')
print('Bad Data : ',baddata)
print('Good Data : ',gooddata)
```

Output:

```
In [10]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py',
wdir='C:/Users/MSCIT/.spyder-py3')


#1 Removing leading or lagging spaces from a data entry
>  Data Science with too many spaces is bad!!!  <
> Data Science with too many spaces is bad!!! <

#2 Removing nonprintable characters from a data entry
Bad Data :  Data Science with¬ funny characters is ┼bad!!!
Clean Data :  DataScience with funny characters is bad!!!

# 3 Reformatting data entry to match specific formatting
criteria.
Bad Data :  2019-10-31
Good Data :  31 October 2019
```

B. Averaging of Data

Code:

```
import pandas as pd
#####################################################################
InputFileName='IP_DATA_CORE.csv'
OutputFileName='Retrieve_Router_Location.csv'
Base='C:/VKHCG'
print('##################################')
print('Working Base :',Base, ' using ')
print('##################################')
sFileName=Base + '/01-Vermeulen/00-RawData/' + InputFileName
print('Loading :',sFileName)
IP_DATA_ALL=pd.read_csv(sFileName,header=0,low_memory=False,
usecols=['Country','Place Name','Latitude','Longitude'], encoding="latin-1")
IP_DATA_ALL.rename(columns={'Place Name': 'Place_Name'}, inplace=True)
AllData=IP_DATA_ALL[['Country', 'Place_Name','Latitude']]
print(AllData)
MeanData=AllData.groupby(['Country', 'Place_Name'])['Latitude'].mean()
print(MeanData)
```

Output:

```
3561      DE     Munich   48.1480

[3562 rows x 3 columns]
Country  Place_Name
DE       Munich        48.143223
GB       London        51.509406
US       New York      40.747044
Name: Latitude, dtype: float64
```

D. Outlier Detection

Code:

```
#####################################################################
# -*- coding: utf-8 -*-
#####################################################################
import pandas as pd
#####################################################################
InputFileName='IP_DATA_CORE.csv'
OutputFileName='Retrieve_Router_Location.csv'
Base='C:/VKHCG'
print('#################################')
print('Working Base :',Base)
print('#################################')
#####################################################################
sFileName=Base + '/01-Vermeulen/00-RawData/' + InputFileName
print('Loading :',sFileName)
IP_DATA_ALL=pd.read_csv(sFileName,header=0,low_memory=False,
usecols=['Country','Place Name','Latitude','Longitude'], encoding="latin-1")
IP_DATA_ALL.rename(columns={'Place Name': 'Place_Name'}, inplace=True)
LondonData=IP_DATA_ALL.loc[IP_DATA_ALL['Place_Name']=='London']
AllData=LondonData[['Country', 'Place_Name','Latitude']]
print('All Data')
print(AllData)
MeanData=AllData.groupby(['Country', 'Place_Name'])['Latitude'].mean()
StdData=AllData.groupby(['Country', 'Place_Name'])['Latitude'].std()
print('Outliers')
UpperBound=float(MeanData+StdData)
print('Higher than ', UpperBound)
OutliersHigher=AllData[AllData.Latitude>UpperBound]
print(OutliersHigher)
LowerBound=float(MeanData-StdData)
print('Lower than ', LowerBound)
OutliersLower=AllData[AllData.Latitude<LowerBound]
print(OutliersLower)
print('Not Outliers')
OutliersNot=AllData[(AllData.Latitude>=LowerBound) & (AllData.Latitude<=UpperBound)]
print(OutliersNot)
```

Output:

```
In [35]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py',
wdir='C:/Users/MSCIT/.spyder-py3')


############################
Working Base : C:/VKHCG
############################
Loading : C:/VKHCG/01-Vermeulen/00-RawData/IP_DATA_CORE.csv
All Data
    Country Place_Name  Latitude
1910      GB     London   51.5130
1911      GB     London   51.5508
1912      GB     London   51.5649
1913      GB     London   51.5895
1914      GB     London   51.5232
1915      GB     London   51.4739
1916      GB     London   51.5491
1917      GB     London   51.5085
```

```
[1502 rows x 3 columns]
Outliers
Higher than  51.51263550786781
     Country Place_Name  Latitude
1910      GB     London   51.5130
1911      GB     London   51.5508
1912      GB     London   51.5649
1913      GB     London   51.5895
1914      GB     London   51.5232
1916      GB     London   51.5491
1919      GB     London   51.5161
1920      GB     London   51.5198
1921      GB     London   51.5198
1923      GB     London   51.5237
1924      GB     London   51.5237
1925      GB     London   51.5237
1926      GB     London   51.5237
1927      GB     London   51.5232
3436      GB     London   51.5163
3438      GB     London   51.5136
Lower than  51.50617687562166
     Country Place_Name  Latitude
1915      GB     London   51.4739
```

```
Not Outliers
     Country Place_Name  Latitude
1917      GB     London   51.5085
1918      GB     London   51.5085
1922      GB     London   51.5085
1928      GB     London   51.5085
1929      GB     London   51.5085
1957      GB     London   51.5115
1958      GB     London   51.5092
1959      GB     London   51.5092
1960      GB     London   51.5092
1961      GB     London   51.5092
1962      GB     London   51.5092
```

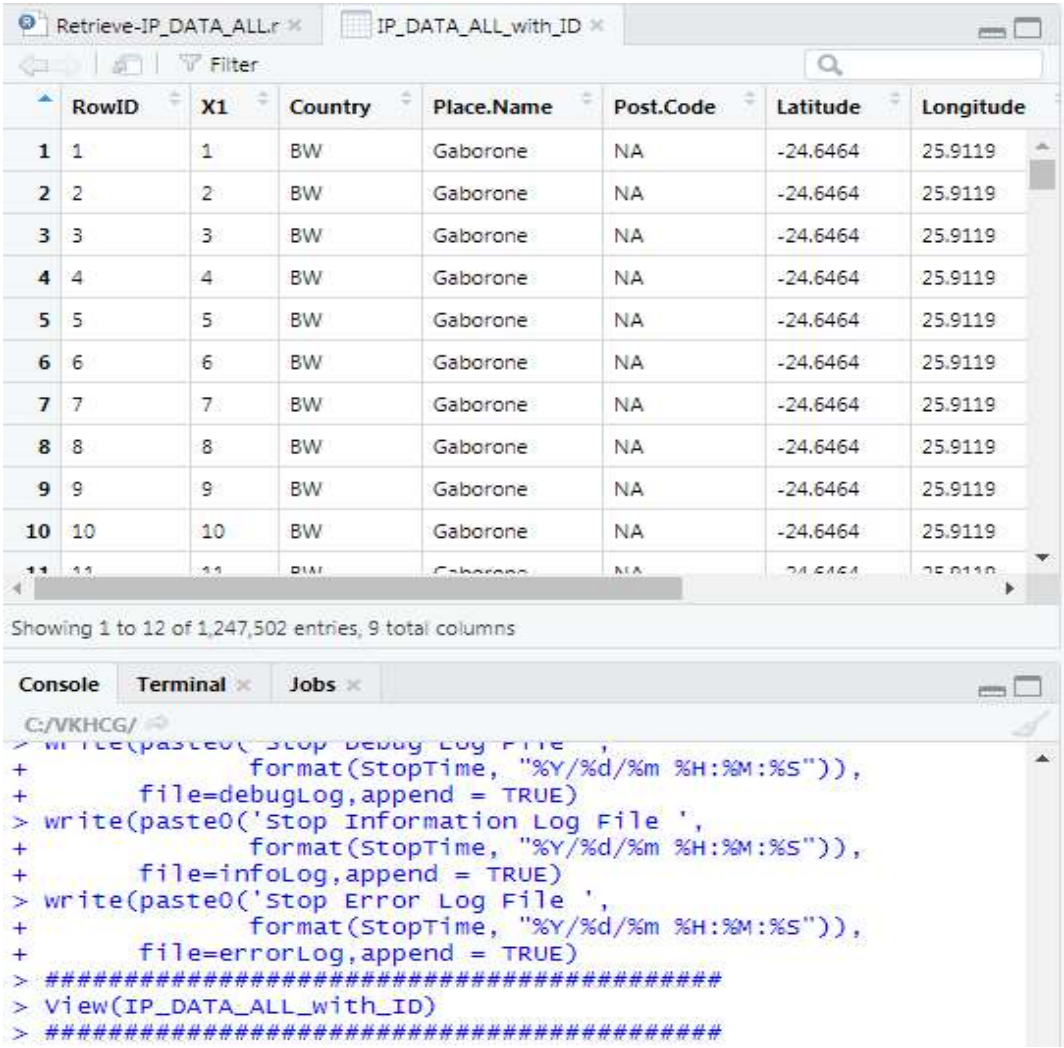
E. Logging

Code:

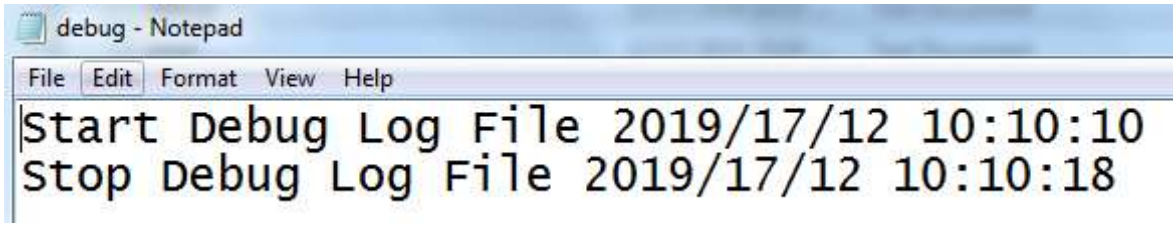
```
import sys
import os
import logging
import uuid
import shutil
    import time
if sys.platform == 'linux':
Base=os.path.expanduser('~') + '/VKHCG'
else:
    Base='C:/VKHCG'
sCompanies=['01-Vermeulen','02-Krennwallner','03-Hillman','04-Clark']
sLayers=['01-Retrieve','02-Assess','03-Process','04-Transform','05-Organise','
06-Report']
    sLevels=['debug','info','warning','error']
for sCompany in sCompanies:
sFileDir=Base + '/' + sCompany
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
for sLayer in sLayers:
log = logging.getLogger() # root logger
for hdlr in log.handlers[:]: # remove all old handlers
log.removeHandler(hdlr)
sFileDir=Base + '/' + sCompany + '/' + sLayer + '/Logging'
if os.path.exists(sFileDir):
shutil.rmtree(sFileDir)
time.sleep(2)
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
skey=str(uuid.uuid4())
sLogFile=Base + '/' + sCompany + '/' + sLayer + '/Logging/
Logging_'+skey+'.log'
print('Set up:',sLogFile)
logging.basicConfig(level=logging.DEBUG,
format='%(asctime)s %(name)-12s %(levelname)-8s %(message)s',
datefmt='%m-%d %H:%M',
filename=sLogFile,
filemode='w')
console = logging.StreamHandler()
console.setLevel(logging.INFO)
formatter = logging.Formatter('%(name)-12s: %(levelname)-8s %(message)s')
console.setFormatter(formatter)
logging.getLogger('').addHandler(console)
```

```
logging.info('Practical Data Science is fun!.')
for sLevel in sLevels:
sApp='Application-'+ sCompany + '-' + sLayer + '-' + sLevel
logger = logging.getLogger(sApp)
if sLevel == 'debug':
logger.debug('Practical Data Science logged a debugging message.')
if sLevel == 'info':
logger.info('Practical Data Science logged information message.')
if sLevel == 'warning':
logger.warning('Practical Data Science logged a warning message.')
if sLevel == 'error':
logger.error('Practical Data Science logged an error message.')
```

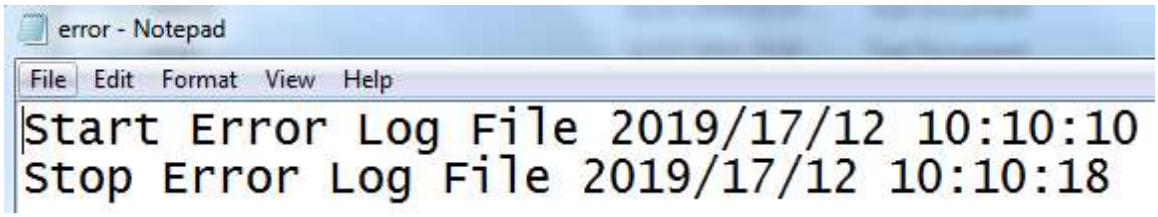
Output:



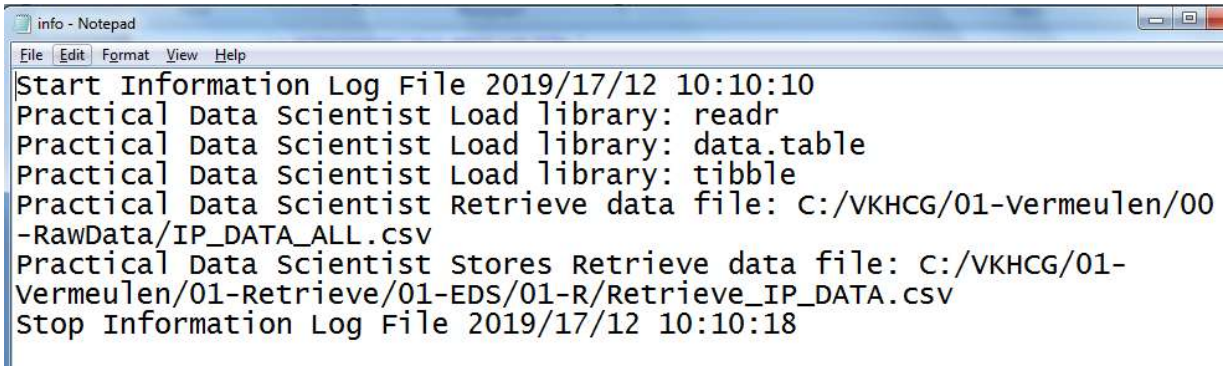
Debug.txt



Error.txt



Info.txt



Practical 4

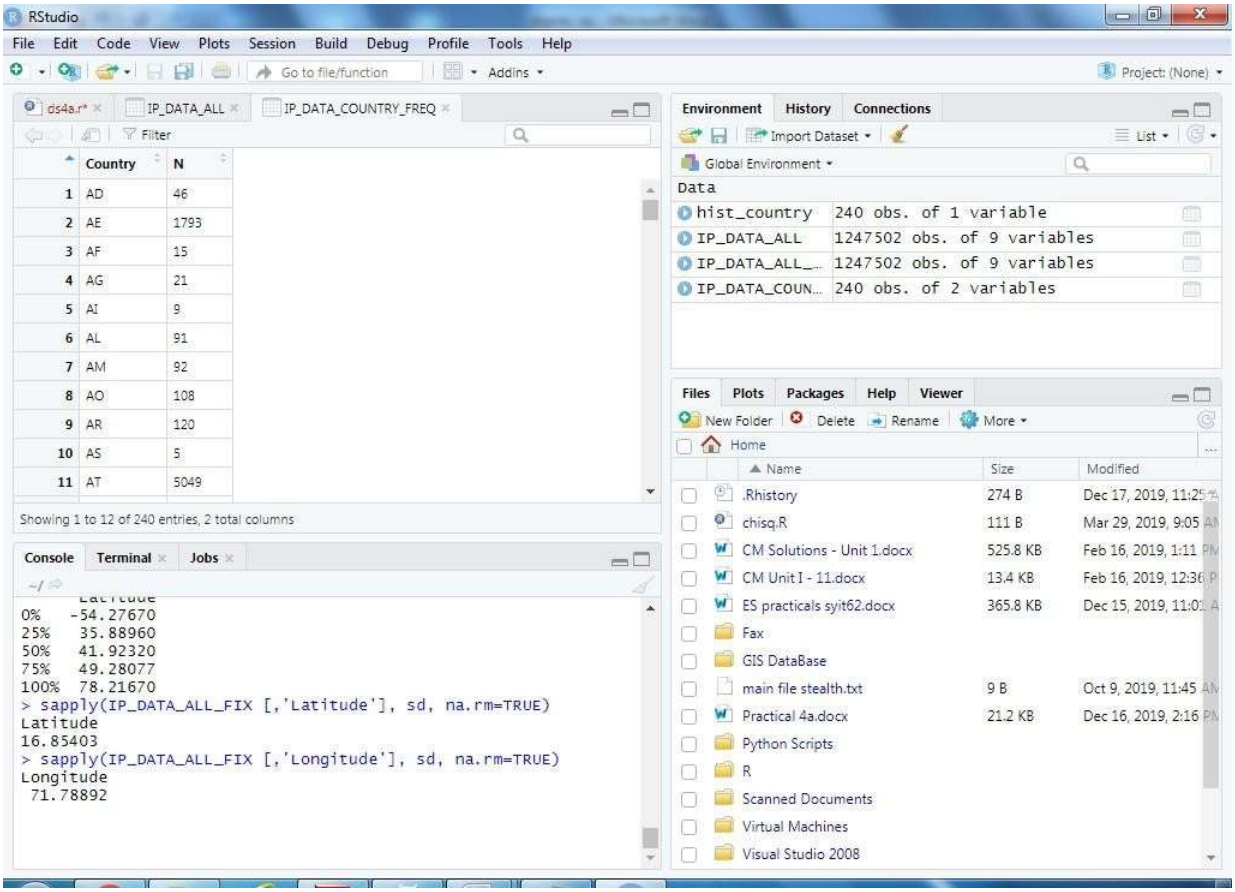
Retrieve Superstep

A. Perform the following data processing using R.

Code:

```
library(readr)
IP_DATA_ALL <- read_csv("C:/VKHCG/01-Vermeulen/00-RawData/IP_DATA_ALL.csv")
View(IP_DATA_ALL)
spec(IP_DATA_ALL)
IP_DATA_ALL_FIX=set_tidy_names(IP_DATA_ALL, syntactic = TRUE, quiet = TRUE)
library(tibble)
set_tidy_names(IP_DATA_ALL, syntactic = TRUE, quiet = FALSE)
sapply(IP_DATA_ALL_FIX, typeof)
library(data.table)
hist_country=data.table(Country=unique(IP_DATA_ALL_FIX[is.na(IP_DATA_ALL_FIX
['Country']) == 0, ]$Country
))
IP_DATA_COUNTRY_FREQ=data.table(with(IP_DATA_ALL_FIX, table(Country)))
View(IP_DATA_COUNTRY_FREQ)
sapply(IP_DATA_ALL_FIX[,'Latitude'], min, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX[,'Country'], min, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX[,'Latitude'], max, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX[,'Country'], max, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Latitude'], mean, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Latitude'], median, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Latitude'], range, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Latitude'], quantile, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Latitude'], sd, na.rm=TRUE)
sapply(IP_DATA_ALL_FIX [,'Longitude'], sd, na.rm=TRUE)
```

Output:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
ds4a.r*
IP_DATA_ALL
IP_DATA_COUNTRY_FREQ
Filter
Country N
1 AD 46
2 AE 1793
3 AF 15
4 AG 21
5 AI 9
6 AL 91
7 AM 92
8 AO 108
9 AR 120
10 AS 5
11 AT 5049
Showing 1 to 12 of 240 entries, 2 total columns
Console Terminal Jobs
0% -54.27670
25% 35.88960
50% 41.92320
75% 49.28077
100% 78.21670
> sapply(IP_DATA_ALL_FIX [,'Latitude'], sd, na.rm=TRUE)
Latitude
16.85403
> sapply(IP_DATA_ALL_FIX [,'Longitude'], sd, na.rm=TRUE)
Longitude
71.78892
Environment History Connections
Import Dataset
List
Global Environment
Data
hist_country 240 obs. of 1 variable
IP_DATA_ALL 1247502 obs. of 9 variables
IP_DATA_ALL_... 1247502 obs. of 9 variables
IP_DATA_COUN... 240 obs. of 2 variables
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home
Name Size Modified
.Rhistory 274 B Dec 17, 2019, 11:25
chisq.R 111 B Mar 29, 2019, 9:05
CM Solutions - Unit 1.docx 525.8 KB Feb 16, 2019, 1:11
CM Unit I - 11.docx 13.4 KB Feb 16, 2019, 12:36
ES practicals syit62.docx 365.8 KB Dec 15, 2019, 11:01
Fax
GIS DataBase
main file stealth.txt 9 B Oct 9, 2019, 11:45
Practical 4a.docx 21.2 KB Dec 16, 2019, 2:16
Python Scripts
R
Scanned Documents
Virtual Machines
Visual Studio 2008

B. Retrive_IP_DATA_ALL.py

Code:

```
import sys
import os
import pandas as pd
#####################################################################
Base='C:/VKHCG'
#####################################################################
sFileName=Base + '/01-Vermeulen/00-RawData/IP_DATA_ALL.csv'
print('Loading :',sFileName)
IP_DATA_ALL=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
#####################################################################
sFileDir=Base + '/01-Vermeulen/01-Retrieve/01-EDS/02-Python'
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
print('Rows:', IP_DATA_ALL.shape[0])
print('Columns:', IP_DATA_ALL.shape[1])
print('### Raw Data Set #########################################')
for i in range(0,len(IP_DATA_ALL.columns)):
print(IP_DATA_ALL.columns[i],type(IP_DATA_ALL.columns[i]))
print('### Fixed Data Set #######################################')
IP_DATA_ALL_FIX=IP_DATA_ALL
for i in range(0,len(IP_DATA_ALL.columns)):
cNameOld=IP_DATA_ALL_FIX.columns[i] + ' '
cNameNew=cNameOld.strip().replace(" ", ".")
IP_DATA_ALL_FIX.columns.values[i] = cNameNew
print(IP_DATA_ALL.columns[i],type(IP_DATA_ALL.columns[i]))
#####################################################################
#print(IP_DATA_ALL_FIX.head())
#####################################################################
print('Fixed Data Set with ID')
IP_DATA_ALL_with_ID=IP_DATA_ALL_FIX
IP_DATA_ALL_with_ID.index.names = ['RowID']
#print(IP_DATA_ALL_with_ID.head())
sFileName2=sFileDir + '/Retrieve_IP_DATA.csv'
IP_DATA_ALL_with_ID.to_csv(sFileName2, index = True, encoding="latin-1")
#####################################################################
print('### Done!! ################################################')
```

Output:

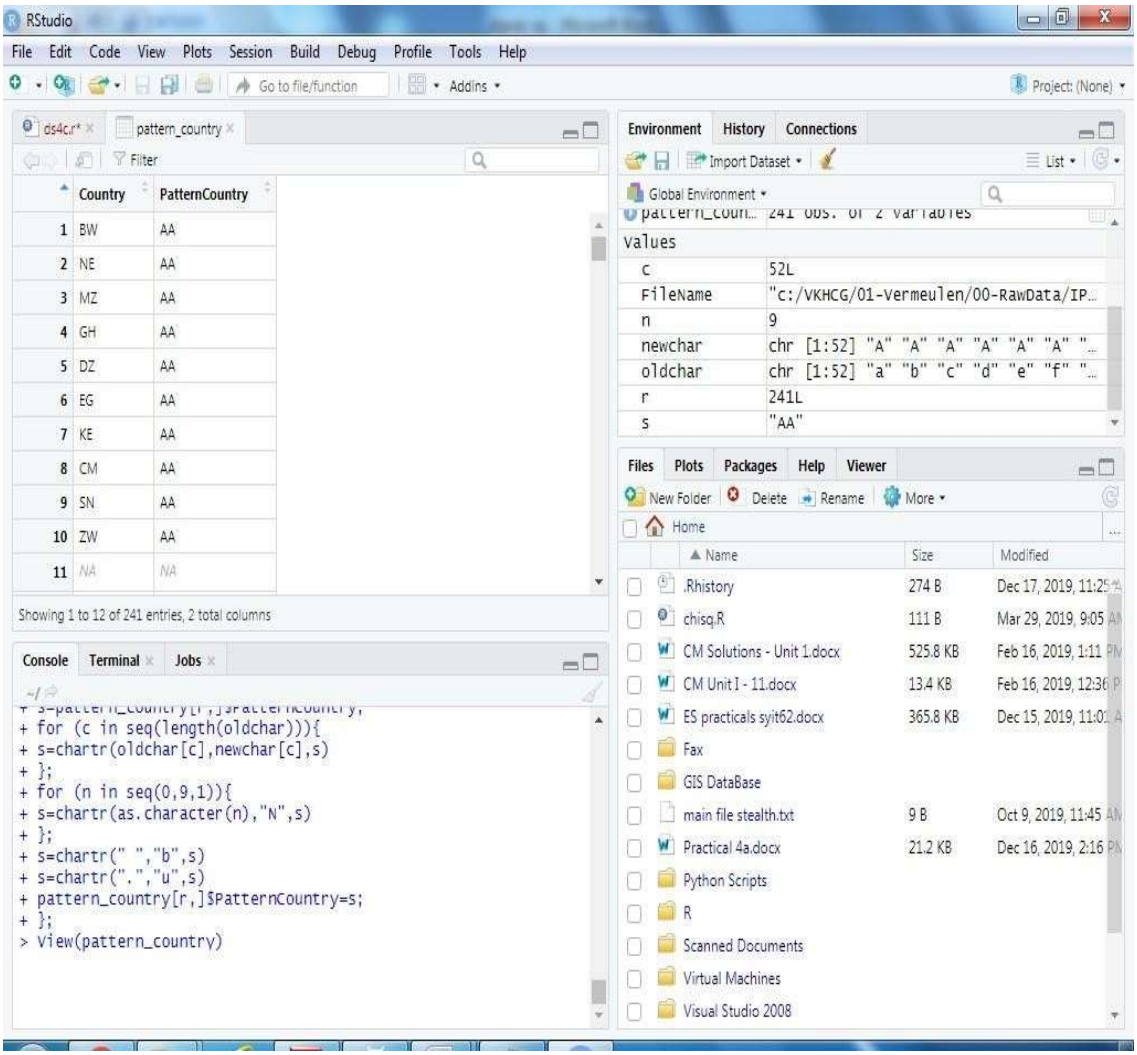
```
In [42]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py',
wdir='C:/Users/MSCIT/.spyder-py3')


Loading : C:/VKHCG/01-Vermeulen/00-RawData/IP_DATA_ALL.csv
### Raw Data Set #####################################
Unnamed: 0 <class 'str'>
ID <class 'str'>
Country <class 'str'>
Place.Name <class 'str'>
Post.Code <class 'str'>
Latitude <class 'str'>
Longitude <class 'str'>
First.IP.Number <class 'str'>
Last.IP.Number <class 'str'>
### Fixed Data Set ###################################
Unnamed:.0 <class 'str'>
ID <class 'str'>
Country <class 'str'>
Place.Name <class 'str'>
Post.Code <class 'str'>
Latitude <class 'str'>
Longitude <class 'str'>
First.IP.Number <class 'str'>
Last.IP.Number <class 'str'>
```

C. Data Pattern

Code:

```
library(readr)
library(data.table)
FileName=paste0('c:/VKHCG/01-Vermeulen/00-RawData/IP_DATA_ALL.csv')
IP_DATA_ALL <- read_csv(FileName)
hist_country=data.table(Country=unique(IP_DATA_ALL$Country))
pattern_country=data.table(Country=hist_country$Country,
PatternCountry=hist_country$Country)
oldchar=c(letters,LETTERS)
newchar=replicate(length(oldchar),"A")
for (r in seq(nrow(pattern_country))){
s=pattern_country[r,]$PatternCountry;
for (c in seq(length(oldchar))){
s=chartr(oldchar[c],newchar[c],s)
};
for (n in seq(0,9,1)){
s=chartr(as.character(n),"N",s)
};
s=chartr(" ","b",s)
s=chartr(".","u",s)
pattern_country[r,]$PatternCountry=s;
};
View(pattern_country)
```

Output:


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
ds4c.r*
pattern_country
Filter
Country PatternCountry
1 BW AA
2 NE AA
3 MZ AA
4 GH AA
5 DZ AA
6 EG AA
7 KE AA
8 CM AA
9 SN AA
10 ZW AA
11 NA NA
Showing 1 to 12 of 241 entries, 2 total columns
Console Terminal Jobs
+ for (c in seq(length(oldchar))){
+ s=chartr(oldchar[c],newchar[c],s)
+ };
+ for (n in seq(0,9,1)){
+ s=chartr(as.character(n),"N",s)
+ };
+ s=chartr(" ","b",s)
+ s=chartr(".","u",s)
+ pattern_country[r,]$PatternCountry=s;
+ };
> View(pattern_country)
Environment History Connections
Import Dataset
List
Global Environment
Values
c 52L
FileName "c:/VKHCG/01-Vermeulen/00-RawData/IP...
n 9
newchar chr [1:52] "A" "A" "A" "A" "A" "A" "...
oldchar chr [1:52] "a" "b" "c" "d" "e" "f" "...
r 241L
s "AA"
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home
Name Size Modified
.Rhistory 274 B Dec 17, 2019, 11:25
chisq.R 111 B Mar 29, 2019, 9:05
CM Solutions - Unit 1.docx 525.8 KB Feb 16, 2019, 1:11
CM Unit I - 11.docx 13.4 KB Feb 16, 2019, 12:36
ES practicals syit62.docx 365.8 KB Dec 15, 2019, 11:01
Fax
GIS DataBase
main file stealth.txt 9 B Oct 9, 2019, 11:45
Practical 4a.docx 21.2 KB Dec 16, 2019, 2:16
Python Scripts
R
Scanned Documents
Virtual Machines
Visual Studio 2008

D. Loading IP_DATA_ALL

Code:

```
##############Retrieve-IP_DATA_ALL.py#########################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
#####################################################################
Base='C:/VKHCG'
#####################################################################
sFileName=Base + '/01-Vermeulen/00-RawData/IP_DATA_ALL.csv'
print('Loading :',sFileName)
IP_DATA_ALL=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
#####################################################################
sFileDir=Base + '/01-Vermeulen/01-Retrieve/01-EDS/02-Python'
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
print('Rows:', IP_DATA_ALL.shape[0])
print('Columns:', IP_DATA_ALL.shape[1])
print('### Raw Data Set #########################################')
for i in range(0,len(IP_DATA_ALL.columns)):
print(IP_DATA_ALL.columns[i],type(IP_DATA_ALL.columns[i]))
print('### Fixed Data Set #########################################')
IP_DATA_ALL_FIX=IP_DATA_ALL
for i in range(0,len(IP_DATA_ALL.columns)):
cNameOld=IP_DATA_ALL_FIX.columns[i] + ' '
cNameNew=cNameOld.strip().replace(" ", ".")
IP_DATA_ALL_FIX.columns.values[i] = cNameNew
print(IP_DATA_ALL.columns[i],type(IP_DATA_ALL.columns[i]))
#####################################################################
#print(IP_DATA_ALL_FIX.head())
#####################################################################
print('Fixed Data Set with ID')
IP_DATA_ALL_with_ID=IP_DATA_ALL_FIX
IP_DATA_ALL_with_ID.index.names = ['RowID']
#print(IP_DATA_ALL_with_ID.head())
sFileName2=sFileDir + '/Retrieve_IP_DATA.csv'
IP_DATA_ALL_with_ID.to_csv(sFileName2, index = True, encoding="latin-1")
#####################################################################
print('### Done!! ############################################')
#####################################################################
```

Output:

```
In [59]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py', wdir='C:/
Users/MSCIT/.spyder-py3')


Loading : C:/VKHCG/01-Vermeulen/00-RawData/COUNTRY-CODES.csv
### Raw Data Set #####################################
Code <class 'str'>
Country name <class 'str'>
Year <class 'str'>
country code top-level domain <class 'str'>
### Fixed Data Set ###################################
Code <class 'str'>
Country.name <class 'str'>
Year <class 'str'>
country.code.top-level.domain <class 'str'>
```

E. Program to connect to different data sources.

SQLite:

Code:

```
########################################## ########################
# -*- coding: utf-8 -*-
#########################################################################
import sqlite3 as sq
import pandas as pd
#########################################################################
Base='C:/VKHCG'
sDatabaseName=Base + '/01-Vermeulen/00-RawData/SQLite/vermeulen.db'
conn = sq.connect(sDatabaseName)
#########################################################################
sFileName='C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retrieve_IP_DATA.csv'
print('Loading :',sFileName)
IP_DATA_ALL_FIX=pd.read_csv(sFileName,header=0,low_memory=False)
IP_DATA_ALL_FIX.index.names = ['RowIDCSV']
sTable='IP_DATA_ALL'
print('Storing :',sDatabaseName,' Table:',sTable)
IP_DATA_ALL_FIX.to_sql(sTable, conn, if_exists="replace")
print('Loading :',sDatabaseName,' Table:',sTable)
TestData=pd.read_sql_query("select * from IP_DATA_ALL;", conn)
print('#################')
print('## Data Values')
print('#################')
print(TestData)
print('#################')
print('## Data Profile')
print('#################')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('#################')
print('### Done!! ############################################')
```

Output:

```
In [9]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py', wdir='C:/Users/
MSCIT/.spyder-py3')

Loading : C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retrieve_IP_DATA.csv
Storing : C:/VKHCG/01-Vermeulen/00-RawData/SQLite/vermeulen.db  Table: IP_DATA_ALL
Loading : C:/VKHCG/01-Vermeulen/00-RawData/SQLite/vermeulen.db  Table: IP_DATA_ALL
Data Values
################
    RowIDCSV  RowID  ID  ... Longitude First.IP.Number Last.IP.Number
0          0      0   1  ...   25.9119       692781056      692781567
1          1      1   2  ...   25.9119       692781824      692783103
2          2      2   3  ...   25.9119       692909056      692909311
3          3      3   4  ...   25.9119       692909568      692910079
4          4      4   5  ...   25.9119       693051392      693052415
5          5      5   6  ...   25.9119       693078272      693078527
6          6      6   7  ...   25.9119       693608448      693616639
7          7      7   8  ...   25.9119       696929792      696930047
8          8      8   9  ...   25.9119       700438784      700439039
9          9      9  10  ...   25.9119       702075904      702076927
10        10     10  11  ...   25.9119       702498816      702499839
11        11     11  12  ...   25.9119       702516224      702517247
12        12     12  13  ...   25.9119       774162663      774162667
13        13     13  14  ...   25.9119      1401887232     1401887743
14        14     14  15  ...   25.9119      1754209024     1754209279
15        15     15  16  ...    2.1167       696918528      696919039
16        16     16  17  ...    2.1167       696922112      696924159
17        17     17  18  ...    2.1167       701203456      701203711
18        18     18  19  ...    2.1167       758886912      758887167
19        19     19  20  ...    2.1167      1347294153     1347294160
20        20     20  21  ...    2.1167      1755100096     1755100351
21        21     21  22  ...    2.1167      1755828480     1755828735
22        22     22  23  ...   32.5892       692883456      692883967
23        23     23  24  ...   32.5892       692944896      692946943
24        24     24  25  ...   32.5892       696967168      696971263
25        25     25  26  ...   32.5892       700358656      700360959
26        26     26  27  ...   32.5892       702292992      702294015
```

Practical 5

Assessing Data

5A1. Drop the Columns Where All Elements Are Missing Values

Code:

```
##################### Assess-Good-Bad-01.py#########################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
#####################################################################
Base='C:/VKHCG'
#####################################################################
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#####################################################################
sInputFileName='Good-or-Bad.csv'
sOutputFileName='Good-or-Bad-01.csv'
Company='01-Vermeulen'
#####################################################################
Base='C:/VKHCG'
#####################################################################
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
#####################################################################
### Import Warehouse
#####################################################################
sFileName=Base + '/' + Company + '/00-RawData/' + sInputFileName
print('Loading :',sFileName)
RawData=pd.read_csv(sFileName,header=0)
print('#################################')
print('## Raw Data Values')
print('#################################')
print(RawData)
print('#################################')
print('## Data Profile')
print('#################################')
print('Rows :',RawData.shape[0])
print('Columns :',RawData.shape[1])
print('#################################')
```

```
#####################################################################
sFileName=sFileDir + '/' + sInputFileName
RawData.to_csv(sFileName, index = False)
#####################################################################
TestData=RawData.dropna(axis=1, how='all')
#####################################################################
print('###################################')
print('## Test Data Values')
print('###################################')
print(TestData)
print('###################################')
print('## Data Profile')
print('###################################')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('###################################')
#####################################################################
sFileName=sFileDir + '/' + sOutputFileName
TestData.to_csv(sFileName, index = False)
#####################################################################
print('###################################')
print('### Done!! ####################')
print('###################################')
```

Output:

```
In [16]: runfile('C:/VKHCG/01-Vermeulen/02-Assess/Assess-Good-
Bad-01.py', wdir='C:/VKHCG/01-Vermeulen/02-Assess')


################################
Working Base : C:/VKHCG  using  win32
################################
Loading : C:/VKHCG/01-Vermeulen/00-RawData/Good-or-Bad.csv
################################
## Raw Data Values
################################
      ID FieldA  FieldB FieldC  FieldD  FieldE    FieldF  FieldG
0    1.0   Good  Better   Best  1024.0     NaN   10241.0       1
1    2.0   Good     NaN   Best   512.0     NaN    5121.0       2
2    3.0   Good  Better    NaN   256.0     NaN     256.0       3
3    4.0   Good  Better   Best     NaN     NaN     211.0       4
4    5.0   Good  Better    NaN    64.0     NaN    6411.0       5
5    6.0   Good     NaN   Best    32.0     NaN      32.0       6
6    7.0    NaN  Better   Best    16.0     NaN    1611.0       7
7    8.0    NaN     NaN   Best     8.0     NaN    8111.0       8
8    9.0    NaN     NaN    NaN     4.0     NaN      41.0       9
9   10.0      A       B      C     2.0     NaN   21111.0      10
10   NaN    NaN     NaN    NaN     NaN     NaN       NaN      11
11  10.0   Good  Better   Best  1024.0     NaN  102411.0      12
12  10.0   Good     NaN   Best   512.0     NaN     512.0      13
13  10.0   Good  Better    NaN   256.0     NaN    1256.0      14
14  10.0   Good  Better   Best     NaN     NaN       NaN      15
15  10.0   Good  Better    NaN    64.0     NaN     164.0      16
16  10.0   Good     NaN   Best    32.0     NaN     322.0      17
17  10.0    NaN  Better   Best    16.0     NaN     163.0      18
18  10.0    NaN     NaN   Best     8.0     NaN     844.0      19
19  10.0    NaN     NaN    NaN     4.0     NaN    4555.0      20
20  10.0      A       B      C     2.0     NaN     111.0      21
################################
## Data Profile
################################
Rows : 21
Columns : 8
################################
## Test Data Values
################################
      ID FieldA  FieldB FieldC  FieldD    FieldF  FieldG
0    1.0   Good  Better   Best  1024.0   10241.0       1
1    2.0   Good     NaN   Best   512.0    5121.0       2
2    3.0   Good  Better    NaN   256.0     256.0       3
3    4.0   Good  Better   Best     NaN     211.0       4
4    5.0   Good  Better    NaN    64.0    6411.0       5
5    6.0   Good     NaN   Best    32.0      32.0       6
6    7.0    NaN  Better   Best    16.0    1611.0       7
7    8.0    NaN     NaN   Best     8.0    8111.0       8
8    9.0    NaN     NaN    NaN     4.0      41.0       9
9   10.0      A       B      C     2.0   21111.0      10
10   NaN    NaN     NaN    NaN     NaN       NaN      11
11  10.0   Good  Better   Best  1024.0  102411.0      12
12  10.0   Good     NaN   Best   512.0     512.0      13
13  10.0   Good  Better    NaN   256.0    1256.0      14
14  10.0   Good  Better   Best     NaN       NaN      15
15  10.0   Good  Better    NaN    64.0     164.0      16
16  10.0   Good     NaN   Best    32.0     322.0      17
17  10.0    NaN  Better   Best    16.0     163.0      18
18  10.0    NaN     NaN   Best     8.0     844.0      19
19  10.0    NaN     NaN    NaN     4.0    4555.0      20
20  10.0      A       B      C     2.0     111.0      21
################################
## Data Profile
################################
Rows : 21
Columns : 7
################################
Done!!
```

5A2. Drop the Columns Where Any of the Elements Is Missing Values

Code:

```
################### Assess-Good-Bad-02.py#############################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
#####################################################################
Base='C:/VKHCG'
sInputFileName='Good-or-Bad.csv'
sOutputFileName='Good-or-Bad-02.csv'
Company='01-Vermeulen'
#####################################################################
Base='C:/VKHCG'
#####################################################################
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#####################################################################
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
#####################################################################
### Import Warehouse
#####################################################################
sFileName=Base + '/' + Company + '/00-RawData/' + sInputFileName
print('Loading :',sFileName)
RawData=pd.read_csv(sFileName,header=0)
print('#################################')
print('## Raw Data Values')
print('#################################')
print(RawData)
print('#################################')
print('## Data Profile')
print('#################################')
print('Rows :',RawData.shape[0])
print('Columns :',RawData.shape[1])
print('#################################')
#####################################################################
sFileName=sFileDir + '/' + sInputFileName
RawData.to_csv(sFileName, index = False)
#####################################################################
TestData=RawData.dropna(axis=1, how='any')
```

```
#####################################################################
print('###################################')
print('## Test Data Values')
print('###################################')
print(TestData)
print('###################################')
print('## Data Profile')
print('###################################')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('###################################')
#####################################################################
sFileName=sFileDir + '/' + sOutputFileName
TestData.to_csv(sFileName, index = False)
#####################################################################
print('###################################')
print('### Done!! ####################')
print('###################################')
```

Output:

```
In [22]: runfile('C:/VKHCG/01-Vermeulen/02-Assess/Assess-Good-
Bad-02.py', wdir='C:/VKHCG/01-Vermeulen/02-Assess')

################################
Working Base : C:/VKHCG  using  win32
################################
Loading : C:/VKHCG/01-Vermeulen/00-RawData/Good-or-Bad.csv
################################
## Raw Data Values
################################
      ID FieldA  FieldB FieldC  FieldD  FieldE    FieldF  FieldG
0    1.0   Good  Better   Best  1024.0     NaN   10241.0       1
1    2.0   Good     NaN   Best   512.0     NaN    5121.0       2
2    3.0   Good  Better    NaN   256.0     NaN     256.0       3
3    4.0   Good  Better   Best     NaN     NaN     211.0       4
4    5.0   Good  Better    NaN    64.0     NaN    6411.0       5
5    6.0   Good     NaN   Best    32.0     NaN      32.0       6
6    7.0    NaN  Better   Best    16.0     NaN    1611.0       7
7    8.0    NaN     NaN   Best     8.0     NaN    8111.0       8
8    9.0    NaN     NaN    NaN     4.0     NaN      41.0       9
9   10.0      A       B      C     2.0     NaN   21111.0      10
10   NaN    NaN     NaN    NaN     NaN     NaN       NaN      11
11  10.0   Good  Better   Best  1024.0     NaN  102411.0      12
12  10.0   Good     NaN   Best   512.0     NaN     512.0      13
13  10.0   Good  Better    NaN   256.0     NaN    1256.0      14
14  10.0   Good  Better   Best     NaN     NaN       NaN      15
15  10.0   Good  Better    NaN    64.0     NaN     164.0      16
16  10.0   Good     NaN   Best    32.0     NaN     322.0      17
17  10.0    NaN  Better   Best    16.0     NaN     163.0      18
18  10.0    NaN     NaN   Best     8.0     NaN     844.0      19
19  10.0    NaN     NaN    NaN     4.0     NaN    4555.0      20
20  10.0      A       B      C     2.0     NaN     111.0      21

################################
## Data Profile
################################
Rows : 21
Columns : 8
################################
## Test Data Values
################################
    FieldG
0        1
1        2
2        3
3        4
4        5
5        6
6        7
7        8
8        9
9       10
10      11
11      12
12      13
13      14
14      15
15      16
16      17
17      18
18      19
19      20
20      21
################################
## Data Profile
################################
Rows : 21
Columns : 1
################################
Done!!
```

5A3. Keep Only the Rows That Contain a Maximum of Two Missing Values

Code:

```
###################### Assess-Good-Bad-03.py ################
# -*- coding: utf-8 -*-
#########################################################################
import sys
import os
import pandas as pd
#########################################################################
sInputFileName='Good-or-Bad.csv'
sOutputFileName='Good-or-Bad-03.csv'
Company='01-Vermeulen'
Base='C:/VKHCG'
#########################################################################
print('#################################')
print('Working Base :',Base, ' using Windows ~~~~')
print('#################################')
#########################################################################
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
os.makedirs(sFileDir)
#########################################################################
### Import Warehouse
#########################################################################
sFileName=Base + '/' + Company + '/00-RawData/' + sInputFileName
print('Loading :',sFileName)
RawData=pd.read_csv(sFileName,header=0)
print('#################################')
print('## Raw Data Values')
print('#################################')
print(RawData)
print('#################################')
print('## Data Profile')
print('#################################')
print('Rows :',RawData.shape[0])
print('Columns :',RawData.shape[1])
print('#################################')
#########################################################################
sFileName=sFileDir + '/' + sInputFileName
RawData.to_csv(sFileName, index = False)
#########################################################################
TestData=RawData.dropna(thresh=2)

print('#################################')
```

```
print('## Test Data Values')
print('###################################')
print(TestData)
print('###################################')
print('## Data Profile')
print('###################################')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('###################################')
sFileName=sFileDir + '/' + sOutputFileName
TestData.to_csv(sFileName, index = False)
#########################################################################
print('###################################')
print('### Done!! ######################')
print('###################################')
```

Output:

```
In [35]: runfile('C:/VKHCG/01-Vermeulen/02-Assess/Assess-Good-
Bad-03.py', wdir='C:/VKHCG/01-Vermeulen/02-Assess')


################################
Working Base : C:/VKHCG  using  win32
Loading : C:/VKHCG/01-Vermeulen/00-RawData/Good-or-Bad.csv
################################
## Raw Data Values
################################
      ID FieldA  FieldB FieldC  FieldD  FieldE    FieldF  FieldG
0    1.0   Good  Better   Best  1024.0     NaN   10241.0       1
1    2.0   Good     NaN   Best   512.0     NaN    5121.0       2
2    3.0   Good  Better    NaN   256.0     NaN     256.0       3
3    4.0   Good  Better   Best     NaN     NaN     211.0       4
4    5.0   Good  Better    NaN    64.0     NaN    6411.0       5
5    6.0   Good     NaN   Best    32.0     NaN      32.0       6
6    7.0    NaN  Better   Best    16.0     NaN    1611.0       7
7    8.0    NaN     NaN   Best     8.0     NaN    8111.0       8
8    9.0    NaN     NaN    NaN     4.0     NaN      41.0       9
9   10.0      A       B      C     2.0     NaN   21111.0      10
10   NaN    NaN     NaN    NaN     NaN     NaN       NaN      11
11  10.0   Good  Better   Best  1024.0     NaN  102411.0      12
12  10.0   Good     NaN   Best   512.0     NaN     512.0      13
13  10.0   Good  Better    NaN   256.0     NaN    1256.0      14
14  10.0   Good  Better   Best     NaN     NaN       NaN      15
15  10.0   Good  Better    NaN    64.0     NaN     164.0      16
16  10.0   Good     NaN   Best    32.0     NaN     322.0      17
17  10.0    NaN  Better   Best    16.0     NaN     163.0      18
18  10.0    NaN     NaN   Best     8.0     NaN     844.0      19
19  10.0    NaN     NaN    NaN     4.0     NaN    4555.0      20
20  10.0      A       B      C     2.0     NaN     111.0      21
################################
## Data Profile
################################
Rows : 21
Columns : 8
```

```
## Test Data Values
############################
     ID FieldA  FieldB FieldC  FieldD  FieldE    FieldF  FieldG
0   1.0   Good  Better   Best  1024.0     NaN   10241.0       1
1   2.0   Good     NaN   Best   512.0     NaN    5121.0       2
2   3.0   Good  Better    NaN   256.0     NaN     256.0       3
3   4.0   Good  Better   Best     NaN     NaN     211.0       4
4   5.0   Good  Better    NaN    64.0     NaN    6411.0       5
5   6.0   Good     NaN   Best    32.0     NaN      32.0       6
6   7.0    NaN  Better   Best    16.0     NaN    1611.0       7
7   8.0    NaN     NaN   Best     8.0     NaN    8111.0       8
8   9.0    NaN     NaN    NaN     4.0     NaN      41.0       9
9  10.0      A       B      C     2.0     NaN   21111.0      10
11 10.0   Good  Better   Best  1024.0     NaN  102411.0      12
12 10.0   Good     NaN   Best   512.0     NaN     512.0      13
13 10.0   Good  Better    NaN   256.0     NaN    1256.0      14
14 10.0   Good  Better   Best     NaN     NaN       NaN      15
15 10.0   Good  Better    NaN    64.0     NaN     164.0      16
16 10.0   Good     NaN   Best    32.0     NaN     322.0      17
17 10.0    NaN  Better   Best    16.0     NaN     163.0      18
18 10.0    NaN     NaN   Best     8.0     NaN     844.0      19
19 10.0    NaN     NaN    NaN     4.0     NaN    4555.0      20
20 10.0      A       B      C     2.0     NaN     111.0      21
############################
## Data Profile
############################
Rows : 20
Columns : 8
############################
Done!!
```

5A4. Fill All Missing Values with the Mean, Median, Mode, Minimum, and Maximum of the Particular Numeric Column

Code:

```
 # -*- coding: utf-8 -
*###################################################################### Import sys
import os
 import pandas as pd
######################################################################
Base='C:/VKHCG'
 sInputFileName='Good-or-Bad.csv'
sOutputFileNameA='Good-or-Bad-04-A.csv'
sOutputFileNameB='Good-or-Bad-04-B.csv'
sOutputFileNameC='Good-or-Bad-04-C.csv'
sOutputFileNameD='Good-or-Bad-04-D.csv'
sOutputFileNameE='Good-or-Bad-04-E.csv'
Company='01-Vermeulen'
######################################################################
if sys.platform == 'linux':
Base=os.path.expanduser('~') + '/VKHCG'
else:
Base='C:/VKHCG'
 print('##################################')
print('Working Base:',Base,'using',sys.platform)
print('##################################')
######################################################################
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
        os.makedirs(sFileDir)
######################################################################
 ### Import Warehouse
######################################################################
sFileName=Base + '/' + Company + '/00-RawData/' + sInputFileName
print('Loading:',sFileName)
RawData=pd.read_csv(sFileName,header=0)
print('##################################')
print('## Raw Data Values')
print('##################################')
print(RawData) print('##################################')
 print('## Data Profile')
 print('##################################')
print('Rows:',RawData.shape[0])
print('Columns:',RawData.shape[1])
print('##################################')
######################################################################
sFileName=sFileDir + '/' + sInputFileName
```

```
RawData.to_csv(sFileName,index=False)
##########################################################################
TestData=RawData.fillna(RawData.mean())
##########################################################################
print('################################')
print('## Test Data Values- Mean')
print('################################')
print(TestData)
print('################################')
print('## Data Profile')
print('################################')
print('Rows:',TestData.shape[0])
print('Columns:',TestData.shape[1])
print('################################')
##########################################################################
sFileName=sFileDir + '/' + sOutputFileNameA
TestData.to_csv(sFileName,index=False)

##########################################################################
## This is the important action! The rest of this code snippet
## Only supports this action.
##########################################################################
TestData=RawData.fillna(RawData.median())
##########################################################################
print('################################')
print('## Test Data Values - Median')
print('################################')
print(TestData)
print('################################')
print('## Data Profile')
print('################################')
print('Rows:',TestData.shape[0])
print('Columns:',TestData.shape[1])
print('################################')
##########################################################################
sFileName=sFileDir + '/' + sOutputFileNameB
TestData.to_csv(sFileName,index=False)
##########################################################################
##########################################################################
TestData=RawData.fillna(RawData.mode())
##########################################################################
print('################################')
print('## Test Data Values - Mode')
print('################################')
print(TestData)
print('################################')
```

```
print('## Data Profile')
print('###################################')
print('Rows:',TestData.shape[0])
print('Columns:',TestData.shape[1])
print('###################################')
#####################################################################
sFileName=sFileDir + '/' + sOutputFileNameC
TestData.to_csv(sFileName,index=False)
#####################################################################
#####################################################################
TestData=RawData.fillna(RawData.min())
#####################################################################
print('###################################')
print('## Test Data Values - Minumum')
print('###################################')
print(TestData) print('###################################')
print('## Data Profile')
print('###################################')
print('Rows:',TestData.shape[0])
print('Columns:',TestData.shape[1])
print('###################################')
#####################################################################
sFileName=sFileDir + '/' + sOutputFileNameD
TestData.to_csv(sFileName,index=False)
#####################################################################
#####################################################################
TestData=RawData.fillna(RawData.max())
#####################################################################
print('###################################')
print('## Test Data Values - Maximum')
print('###################################')
print(TestData)
print('###################################')
print('##DataProfile')
print('###################################')
print('Rows:',TestData.shape[0])
print('Columns:',TestData.shape[1])
print('###################################')
#####################################################################
sFileName=sFileDir + '/' + sOutputFileNameE
TestData.to_csv(sFileName,index=False)
#####################################################################
#####################################################################
print('###################################')
print('### Done!! ######################')
```

Output:

```
In [7]: runfile('C:/VKHCG/01-Vermeulen/02-Assess/Assess-Good-Bad-05.py',
wdir='C:/VKHCG/01-Vermeulen/02-Assess')

################################
Working Base : C:/VKHCG  using  win32
################################
Loading : C:/VKHCG/01-Vermeulen/00-RawData/Good-or-Bad.csv
################################
## Raw Data Values
################################
      ID FieldA  FieldB FieldC  FieldD  FieldE    FieldF  FieldG
0    1.0   Good  Better   Best  1024.0     NaN   10241.0       1
1    2.0   Good     NaN   Best   512.0     NaN    5121.0       2
2    3.0   Good  Better    NaN   256.0     NaN     256.0       3
3    4.0   Good  Better   Best     NaN     NaN     211.0       4
4    5.0   Good  Better    NaN    64.0     NaN    6411.0       5
5    6.0   Good     NaN   Best    32.0     NaN      32.0       6
6    7.0    NaN  Better   Best    16.0     NaN    1611.0       7
7    8.0    NaN     NaN   Best     8.0     NaN    8111.0       8
8    9.0    NaN     NaN    NaN     4.0     NaN      41.0       9
9   10.0      A       B      C     2.0     NaN   21111.0      10
10   NaN    NaN     NaN    NaN     NaN     NaN       NaN      11
11  10.0   Good  Better   Best  1024.0     NaN  102411.0      12
12  10.0   Good     NaN   Best   512.0     NaN     512.0      13
13  10.0   Good  Better    NaN   256.0     NaN    1256.0      14
14  10.0   Good  Better   Best     NaN     NaN       NaN      15
15  10.0   Good  Better    NaN    64.0     NaN     164.0      16
16  10.0   Good     NaN   Best    32.0     NaN     322.0      17
17  10.0    NaN  Better   Best    16.0     NaN     163.0      18
18  10.0    NaN     NaN   Best     8.0     NaN     844.0      19
19  10.0    NaN     NaN    NaN     4.0     NaN    4555.0      20
20  10.0      A       B      C     2.0     NaN     111.0      21
################################
## Data Profile
```

Practical 6
Processing Data
Hubs, Links, Satellite

```
#####################################################################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
from datetime import datetime
from datetime import timedelta
from pytz import timezone, all_timezones
import pandas as pd
import sqlite3 as sq
from pandas.io import sql
import uuid

pd.options.mode.chained_assignment = None
#####################################################################
if sys.platform == 'linux':
    Base=os.path.expanduser('~') + '/VKHCG'
else:
    Base='C:/VKHCG'
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#####################################################################
Company='01-Vermeulen'
InputDir='00-RawData'
InputFileName='VehicleData.csv'
#####################################################################
sDataBaseDir=Base + '/' + Company + '/03-Process/SQLite'
if not os.path.exists(sDataBaseDir):
    os.makedirs(sDataBaseDir)
#####################################################################
sDatabaseName=sDataBaseDir + '/Hillman.db'
conn1 = sq.connect(sDatabaseName)
#####################################################################
sDataVaultDir=Base + '/88-DV'
if not os.path.exists(sDataBaseDir):
    os.makedirs(sDataBaseDir)
#####################################################################
sDatabaseName=sDataVaultDir + '/datavault.db'
conn2 = sq.connect(sDatabaseName)
#####################################################################
```

```
base = datetime(2018,1,1,0,0,0)
numUnits=10*365*24
#####################################################################
date_list = [base - timedelta(hours=x) for x in range(0, numUnits)]
t=0
for i in date_list:
   now_utc=i.replace(tzinfo=timezone('UTC'))
   sDateTime=now_utc.strftime("%Y-%m-%d %H:%M:%S")
   print(sDateTime)
   sDateTimeKey=sDateTime.replace(' ','-').replace(':','-')
   t+=1
   IDNumber=str(uuid.uuid4())
   TimeLine=[('ZoneBaseKey', ['UTC']),
         ('IDNumber', [IDNumber]),
         ('nDateTimeValue', [now_utc]),
         ('DateTimeValue', [sDateTime]),
         ('DateTimeKey', [sDateTimeKey])]
   if t==1:
     TimeFrame = pd.DataFrame.from_items(TimeLine)
   else:
      TimeRow = pd.DataFrame.from_items(TimeLine)
      TimeFrame = TimeFrame.append(TimeRow)
#####################################################################
TimeHub=TimeFrame[['IDNumber','ZoneBaseKey','DateTimeKey','DateTimeValue']]
TimeHubIndex=TimeHub.set_index(['IDNumber'],inplace=False)
#####################################################################
TimeFrame.set_index(['IDNumber'],inplace=True)
#####################################################################
sTable = 'Process-Time'
print('Storing :',sDatabaseName,' Table:',sTable)
TimeHubIndex.to_sql(sTable, conn1, if_exists="replace")
#####################################################################
sTable = 'Hub-Time'
print('Storing :',sDatabaseName,' Table:',sTable)
TimeHubIndex.to_sql(sTable, conn2, if_exists="replace")
#####################################################################
active_timezones=all_timezones
z=0
for zone in active_timezones:
   t=0
   for j in range(TimeFrame.shape[0]):
      now_date=TimeFrame['nDateTimeValue'][j]
      DateTimeKey=TimeFrame['DateTimeKey'][j]
      now_utc=now_date.replace(tzinfo=timezone('UTC'))
      sDateTime=now_utc.strftime("%Y-%m-%d %H:%M:%S")
      now_zone = now_utc.astimezone(timezone(zone))
```

```
        sZoneDateTime=now_zone.strftime("%Y-%m-%d %H:%M:%S")
        print(sZoneDateTime)
        t+=1
        z+=1
        IDZoneNumber=str(uuid.uuid4())
        TimeZoneLine=[('ZoneBaseKey', ['UTC']),
                      ('IDZoneNumber', [IDZoneNumber]),
                      ('DateTimeKey', [DateTimeKey]),
                      ('UTCDateTimeValue', [sDateTime]),
                      ('Zone', [zone]),
                      ('DateTimeValue', [sZoneDateTime])]
        if t==1:
          TimeZoneFrame = pd.DataFrame.from_items(TimeZoneLine)
        else:
           TimeZoneRow = pd.DataFrame.from_items(TimeZoneLine)
           TimeZoneFrame = TimeZoneFrame.append(TimeZoneRow)

    TimeZoneFrameIndex=TimeZoneFrame.set_index(['IDZoneNumber'],inplace=False)
    sZone=zone.replace('/','-').replace(' ','')
    ######################################################################
    sTable = 'Process-Time-'+sZone
    print('Storing :',sDatabaseName,' Table:',sTable)
    TimeZoneFrameIndex.to_sql(sTable, conn1, if_exists="replace")
#########################################################################
    ######################################################################
    sTable = 'Satellite-Time-'+sZone
    print('Storing :',sDatabaseName,' Table:',sTable)
    TimeZoneFrameIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('#################')
print('Vacuum Databases')
sSQL="VACUUM;"
sql.execute(sSQL,conn1)
sql.execute(sSQL,conn2)
print('#################')
#########################################################################
print('### Done!! ############################################')
#########################################################################
```

Output:

```
In [33]: runfile('C:/Users/MSCIT/.spyder-py3/temp.py',
wdir='C:/Users/MSCIT/.spyder-py3')


##################################
Working Base : C:/VKHCG  using  win32
##################################
2018-01-01 00:00:00
2017-12-31 23:00:00
2017-12-31 22:00:00
2017-12-31 21:00:00
2017-12-31 20:00:00
2017-12-31 19:00:00
2017-12-31 18:00:00
2017-12-31 17:00:00
2017-12-31 16:00:00
2017-12-31 15:00:00
Storing : C:/VKHCG/88-DV/datavault.db  Table: Process-Time

Storing : C:/VKHCG/88-DV/datavault.db  Table: Hub-Time
2018-01-01 00:00:00
2017-12-31 23:00:00
2017-12-31 22:00:00
2017-12-31 21:00:00
2017-12-31 20:00:00
2017-12-31 19:00:00
2017-12-31 18:00:00
2017-12-31 17:00:00
2017-12-31 16:00:00
2017-12-31 15:00:00
Storing : C:/VKHCG/88-DV/datavault.db  Table: Process-Time-
Africa-Abidjan

Storing : C:/VKHCG/88-DV/datavault.db  Table: Satellite-
Time-Africa-Abidjan
2018-01-01 00:00:00
2017-12-31 23:00:00
2017-12-31 22:00:00
2017-12-31 21:00:00
2017-12-31 20:00:00
2017-12-31 19:00:00
2017-12-31 18:00:00
2017-12-31 17:00:00
2017-12-31 16:00:00
2017-12-31 15:00:00
```

Practical 7
Transforming Data
Sun Model

```
#########################################################################
# -*- coding: utf-8 -*-
#########################################################################
import sys
import os
from datetime import datetime
from pytz import timezone
import pandas as pd
import sqlite3 as sq
import uuid
pd.options.mode.chained_assignment = None
#########################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#########################################################################
Company='01-Vermeulen'
#########################################################################
sDataBaseDir=Base + '/' + Company + '/04-Transform/SQLite'
if not os.path.exists(sDataBaseDir):
   os.makedirs(sDataBaseDir)
#########################################################################
sDatabaseName=sDataBaseDir + '/Vermeulen.db'
conn1 = sq.connect(sDatabaseName)
#########################################################################
sDataWarehousetDir=Base + '/99-DW'
if not os.path.exists(sDataWarehousetDir):
   os.makedirs(sDataWarehousetDir)
#########################################################################
sDatabaseName=sDataWarehousetDir + '/datawarehouse.db'
conn2 = sq.connect(sDatabaseName)
#########################################################################
print('\n#################################')
print('Time Dimension')
BirthZone = 'Atlantic/Reykjavik'
BirthDateUTC = datetime(1960,12,20,10,15,0)
BirthDateZoneUTC=BirthDateUTC.replace(tzinfo=timezone('UTC'))
BirthDateZoneStr=BirthDateZoneUTC.strftime("%Y-%m-%d %H:%M:%S")
BirthDateZoneUTCStr=BirthDateZoneUTC.strftime("%Y-%m-%d %H:%M:%S (%Z) (%z)")
```

```
BirthDate = BirthDateZoneUTC.astimezone(timezone(BirthZone))
BirthDateStr=BirthDate.strftime("%Y-%m-%d %H:%M:%S (%Z) (%z)")
BirthDateLocal=BirthDate.strftime("%Y-%m-%d %H:%M:%S")
#########################################################################
IDTimeNumber=str(uuid.uuid4())
TimeLine=[('TimeID', [IDTimeNumber]),
        ('UTCDate', [BirthDateZoneStr]),
        ('LocalTime', [BirthDateLocal]),
        ('TimeZone', [BirthZone])]
TimeFrame = pd.DataFrame.from_items(TimeLine)
#########################################################################
DimTime=TimeFrame
DimTimeIndex=DimTime.set_index(['TimeID'],inplace=False)
#########################################################################
sTable = 'Dim-Time'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
DimTimeIndex.to_sql(sTable, conn1, if_exists="replace")
DimTimeIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('\n###################################')
print('Dimension Person')
print('\n###################################')
FirstName = 'Guðmundur'
LastName = 'Gunnarsson'
#########################################################################
IDPersonNumber=str(uuid.uuid4())
PersonLine=[('PersonID', [IDPersonNumber]),
            ('FirstName', [FirstName]),
            ('LastName', [LastName]),
            ('Zone', ['UTC']),
            ('DateTimeValue', [BirthDateZoneStr])]
PersonFrame = pd.DataFrame.from_items(PersonLine)
#########################################################################
DimPerson=PersonFrame
DimPersonIndex=DimPerson.set_index(['PersonID'],inplace=False)
#########################################################################
sTable = 'Dim-Person'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
DimPersonIndex.to_sql(sTable, conn1, if_exists="replace")
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('\n###################################')
```

```
print('Fact - Person - time')
print('\n###################################')
IDFactNumber=str(uuid.uuid4())
PersonTimeLine=[('IDNumber',  [IDFactNumber]),
                ('IDPersonNumber', [IDPersonNumber]),
                ('IDTimeNumber', [IDTimeNumber])]
PersonTimeFrame = pd.DataFrame.from_items(PersonTimeLine)
#########################################################################
FctPersonTime=PersonTimeFrame
FctPersonTimeIndex=FctPersonTime.set_index(['IDNumber'],inplace=False)
#########################################################################
sTable = 'Fact-Person-Time'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
FctPersonTimeIndex.to_sql(sTable, conn1, if_exists="replace")
FctPersonTimeIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
```

Output:

```
In [24]: runfile('C:/VKHCG/01-Vermeulen/04-Transform/Transform-
Gunnarsson-Sun-Model.py', wdir='C:/VKHCG/01-Vermeulen/04-
Transform')


##################################
Working Base : C:/VKHCG  using  win32
##################################

##################################
Time Dimension

##################################
Storing : C:/VKHCG/99-DW/datawarehouse.db
 Table: Dim-Time

##################################
C:/VKHCG/01-Vermeulen/04-Transform/Transform-Gunnarsson-Sun-
Model.py:55: FutureWarning: from_items is deprecated. Please use
DataFrame.from_dict(dict(items), ...) instead.
DataFrame.from_dict(OrderedDict(items)) may be used to preserve
the key order.
  DimTime=TimeFrame

##################################
Dimension Person

##################################

##################################
Storing : C:/VKHCG/99-DW/datawarehouse.db
 Table: Dim-Person
##################################
C:/VKHCG/01-Vermeulen/04-Transform/Transform-Gunnarsson-Sun-
Model.py:79: FutureWarning: from_items is deprecated. Please use
DataFrame.from_dict(dict(items), ...) instead.
DataFrame.from_dict(OrderedDict(items)) may be used to preserve
the key order.
  DimPerson=PersonFrame

##################################
Fact - Person - time

##################################

##################################
Storing : C:/VKHCG/99-DW/datawarehouse.db
 Table: Fact-Person-Time

##################################
C:/VKHCG/01-Vermeulen/04-Transform/Transform-Gunnarsson-Sun-
Model.py:98: FutureWarning: from_items is deprecated. Please use
DataFrame.from_dict(dict(items), ...) instead.
DataFrame.from_dict(OrderedDict(items)) may be used to preserve
the key order.
  FctPersonTime=PersonTimeFrame
```

Practical 8A
Organizing Data

A. Horizontal Style

```
#####################################################################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
import sqlite3 as sq
#####################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('##################################')
print('Working Base :',Base, ' using ', sys.platform)
print('##################################')
#####################################################################
#####################################################################
Company='01-Vermeulen'
#####################################################################
sDataWarehouseDir=Base + '/99-DW'
if not os.path.exists(sDataWarehouseDir):
   os.makedirs(sDataWarehouseDir)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datawarehouse.db'
conn1 = sq.connect(sDatabaseName)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datamart.db'
conn2 = sq.connect(sDatabaseName)
#####################################################################
print('#################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT * FROM [Dim-BMI];"
PersonFrame0=pd.read_sql_query(sSQL, conn1)
#####################################################################
print('##################################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
print('##################################')
sSQL="SELECT PersonID,\
      Height,\
      Weight,\
```

```
    bmi,\
    Indicator\
  FROM [Dim-BMI]\
  WHERE \
  Height > 1.5 \
  and Indicator = 1\
  ORDER BY  \
    Height,\
    Weight;"
PersonFrame1=pd.read_sql_query(sSQL, conn1)
#########################################################################
DimPerson=PersonFrame1
DimPersonIndex=DimPerson.set_index(['PersonID'],inplace=False)
#########################################################################
sTable = 'Dim-BMI-Horizontal'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('###################################')
sTable = 'Dim-BMI-Horizontal'
print('Loading :',sDatabaseName,' Table:',sTable)
print('###################################')
sSQL="SELECT * FROM [Dim-BMI];"
PersonFrame2=pd.read_sql_query(sSQL, conn2)
#########################################################################
print('###################################')
print('Full Data Set (Rows):', PersonFrame0.shape[0])
print('Full Data Set (Columns):', PersonFrame0.shape[1])
print('###################################')
print('Horizontal Data Set (Rows):', PersonFrame2.shape[0])
print('Horizontal Data Set (Columns):', PersonFrame2.shape[1])
print('###################################')
#########################################################################
```

Output:

```
In [20]: runfile('C:/VKHCG/01-Vermeulen/05-Organise/Organize-Horizontal.py', wdir='C:/VKHCG/
01-Vermeulen/05-Organise')

################################
Working Base : C:/VKHCG  using  win32
################################
################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################################

################################
Storing : C:/VKHCG/99-DW/datamart.db
 Table: Dim-BMI-Horizontal

################################
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI-Horizontal
################################
################################
Full Data Set (Rows): 1080
Full Data Set (Columns): 5
################################
Horizontal Data Set (Rows): 194
Horizontal Data Set (Columns): 5
################################
```

Practical 8B
Vertical style

```
#####################################################################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
import sqlite3 as sq
#####################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#####################################################################
#####################################################################
Company='01-Vermeulen'
#####################################################################
sDataWarehouseDir=Base + '/99-DW'
if not os.path.exists(sDataWarehouseDir):
   os.makedirs(sDataWarehouseDir)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datawarehouse.db'
conn1 = sq.connect(sDatabaseName)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datamart.db'
conn2 = sq.connect(sDatabaseName)
#####################################################################
print('#################################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT * FROM [Dim-BMI];"
PersonFrame0=pd.read_sql_query(sSQL, conn1)
#####################################################################
print('#################################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
print('#################################')
sSQL="SELECT \
      Height,\
      Weight,\
      Indicator\
  FROM [Dim-BMI];"
PersonFrame1=pd.read_sql_query(sSQL, conn1)
#####################################################################
DimPerson=PersonFrame1
```

```
DimPersonIndex=DimPerson.set_index(['Indicator'],inplace=False)
#########################################################################
sTable = 'Dim-BMI-Vertical'
print('\n#####################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n#####################################')
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('##################')
sTable = 'Dim-BMI-Vertical'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT * FROM [Dim-BMI-Vertical];"
PersonFrame2=pd.read_sql_query(sSQL, conn2)
#########################################################################
print('###################################')
print('Full Data Set (Rows):', PersonFrame0.shape[0])
print('Full Data Set (Columns):', PersonFrame0.shape[1])
print('###################################')
print('Horizontal Data Set (Rows):', PersonFrame2.shape[0])
print('Horizontal Data Set (Columns):', PersonFrame2.shape[1])
print('###################################')
#########################################################################
```

Output:

```
In [20]: runfile('C:/VKHCG/01-Vermeulen/05-Organise/Organize-Horizontal.py', wdir='C:/VKHCG/
01-Vermeulen/05-Organise')


################################
Working Base : C:/VKHCG  using  win32
################################
###############
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################################

################################
Storing : C:/VKHCG/99-DW/datamart.db
 Table: Dim-BMI-Horizontal

################################
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI-Horizontal
################################
################################
Full Data Set (Rows): 1080
Full Data Set (Columns): 5
################################
Horizontal Data Set (Rows): 194
Horizontal Data Set (Columns): 5
################################
```

Practical 8C
Island Style

```
#####################################################################
# -*- coding: utf-8 -*-
#####################################################################
import sys
import os
import pandas as pd
import sqlite3 as sq
#####################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#####################################################################
#####################################################################
Company='01-Vermeulen'
#####################################################################
sDataWarehouseDir=Base + '/99-DW'
if not os.path.exists(sDataWarehouseDir):
   os.makedirs(sDataWarehouseDir)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datawarehouse.db'
conn1 = sq.connect(sDatabaseName)
#####################################################################
sDatabaseName=sDataWarehouseDir + '/datamart.db'
conn2 = sq.connect(sDatabaseName)
#####################################################################
print('################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT * FROM [Dim-BMI];"
PersonFrame0=pd.read_sql_query(sSQL, conn1)
#####################################################################
print('################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT \
      Height,\
      Weight,\
```

```
    Indicator\
  FROM [Dim-BMI]\
  WHERE Indicator > 2\
  ORDER BY  \
    Height,\
    Weight;"
PersonFrame1=pd.read_sql_query(sSQL, conn1)
#########################################################################
DimPerson=PersonFrame1
DimPersonIndex=DimPerson.set_index(['Indicator'],inplace=False)
#########################################################################
sTable = 'Dim-BMI-Vertical'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('###################################')
sTable = 'Dim-BMI-Vertical'
print('Loading :',sDatabaseName,' Table:',sTable)
print('###################################')
sSQL="SELECT * FROM [Dim-BMI-Vertical];"
PersonFrame2=pd.read_sql_query(sSQL, conn2)
#########################################################################
print('###################################')
print('Full Data Set (Rows):', PersonFrame0.shape[0])
print('Full Data Set (Columns):', PersonFrame0.shape[1])
print('###################################')
print('Horizontal Data Set (Rows):', PersonFrame2.shape[0])
print('Horizontal Data Set (Columns):', PersonFrame2.shape[1])
print('###################################')
#########################################################################
```

Output:

```
In [38]: runfile('C:/VKHCG/01-Vermeulen/05-Organise/Organize-
Island.py', wdir='C:/VKHCG/01-Vermeulen/05-Organise')


################################
Working Base : C:/VKHCG  using  win32
################################
################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI

################################
Storing : C:/VKHCG/99-DW/datamart.db
 Table: Dim-BMI-Vertical

################################
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI-Vertical
################################
################################
Full Data Set (Rows): 1080
Full Data Set (Columns): 5
################################
Horizontal Data Set (Rows): 771
Horizontal Data Set (Columns): 3
################################
```

Practical 8D
Secure Vault Style

```
#########################################################################
# -*- coding: utf-8 -*-
#########################################################################
import sys
import os
import pandas as pd
import sqlite3 as sq
#########################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('####################################')
print('Working Base :',Base, ' using ', sys.platform)
print('####################################')
#########################################################################
#########################################################################
Company='01-Vermeulen'
#########################################################################
sDataWarehouseDir=Base + '/99-DW'
if not os.path.exists(sDataWarehouseDir):
   os.makedirs(sDataWarehouseDir)
#########################################################################
sDatabaseName=sDataWarehouseDir + '/datawarehouse.db'
conn1 = sq.connect(sDatabaseName)
#########################################################################
sDatabaseName=sDataWarehouseDir + '/datamart.db'
conn2 = sq.connect(sDatabaseName)
#########################################################################
print('#################')
sTable = 'Dim-BMI'
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT * FROM [Dim-BMI];"
PersonFrame0=pd.read_sql_query(sSQL, conn1)
#########################################################################
print('#################')
sTable = 'Dim-BMI'
```

```
print('Loading :',sDatabaseName,' Table:',sTable)
sSQL="SELECT \
      Height,\
      Weight,\
      Indicator,\
      CASE Indicator\
      WHEN 1 THEN 'Pip'\
      WHEN 2 THEN 'Norman'\
      WHEN 3 THEN 'Grant'\
      ELSE 'Sam'\
      END AS Name\
  FROM [Dim-BMI]\
  WHERE Indicator > 2\
  ORDER BY  \
      Height,\
      Weight;"
PersonFrame1=pd.read_sql_query(sSQL, conn1)
#########################################################################
DimPerson=PersonFrame1
DimPersonIndex=DimPerson.set_index(['Indicator'],inplace=False)
#########################################################################
sTable = 'Dim-BMI-Secure'
print('\n###################################')
print('Storing :',sDatabaseName,'\n Table:',sTable)
print('\n###################################')
DimPersonIndex.to_sql(sTable, conn2, if_exists="replace")
#########################################################################
print('###################################')
sTable = 'Dim-BMI-Secure'
print('Loading :',sDatabaseName,' Table:',sTable)
print('###################################')
sSQL="SELECT * FROM [Dim-BMI-Secure] WHERE Name = 'Sam';"
PersonFrame2=pd.read_sql_query(sSQL, conn2)
#########################################################################
print('###################################')
print('Full Data Set (Rows):', PersonFrame0.shape[0])
print('Full Data Set (Columns):', PersonFrame0.shape[1])
print('###################################')
print('Horizontal Data Set (Rows):', PersonFrame2.shape[0])
print('Horizontal Data Set (Columns):', PersonFrame2.shape[1])
```

```
print('Only Sam Data')
print(PersonFrame2.head())
print('###################################')
#########################################################################
```

Output:

```
In [47]: runfile('C:/VKHCG/01-Vermeulen/05-Organise/Organize-Secure-
Vault.py', wdir='C:/VKHCG/01-Vermeulen/05-Organise')


################################
Working Base : C:/VKHCG  using  win32
################################
################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI
################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI

################################
Storing : C:/VKHCG/99-DW/datamart.db
 Table: Dim-BMI-Secure

################################
################################
Loading : C:/VKHCG/99-DW/datamart.db  Table: Dim-BMI-Secure
################################
################################
Full Data Set (Rows): 1080
Full Data Set (Columns): 5
################################
Horizontal Data Set (Rows): 692
Horizontal Data Set (Columns): 4
Only Sam Data
   Indicator  Height  Weight Name
0          4     1.0      35  Sam
1          4     1.0      40  Sam
2          4     1.0      45  Sam
3          4     1.0      50  Sam
4          4     1.0      55  Sam
################################
```

Practical 9
Reporting Data
A. Create a Network Routing Diagram

```
#########################################################################
import sys
import os
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
#########################################################################
pd.options.mode.chained_assignment = None
#########################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + 'VKHCG'
else:
   Base='C:/VKHCG'
#########################################################################
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#########################################################################
sInputFileName='02-Assess/01-EDS/02-Python/Assess-Network-Routing-Company.csv'
#########################################################################
sOutputFileName1='05-Organise/01-EDS/02-Python/Organise-Network-Routing-Company.gml'
sOutputFileName2='05-Organise/01-EDS/02-Python/Organise-Network-Routing-Company.png'
Company='01-Vermeulen'
#########################################################################
#########################################################################
### Import Country Data
#########################################################################
sFileName=Base + '/' + Company + '/' + sInputFileName
print('#################################')
print('Loading :',sFileName)
print('#################################')
CompanyData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
print('#################################')
#########################################################################
print(CompanyData.head())
print(CompanyData.shape)
```

```
#####################################################################
G=nx.Graph()
for i in range(CompanyData.shape[0]):
   for j in range(CompanyData.shape[0]):
      Node0=CompanyData['Company_Country_Name'][i]
      Node1=CompanyData['Company_Country_Name'][j]
      if Node0 != Node1:
         G.add_edge(Node0,Node1)

for i in range(CompanyData.shape[0]):
   Node0=CompanyData['Company_Country_Name'][i]
   Node1=CompanyData['Company_Place_Name'][i] + '('+
CompanyData['Company_Country_Name'][i] + ')'
   if Node0 != Node1:
      G.add_edge(Node0,Node1)

print('Nodes:', G.number_of_nodes())
print('Edges:', G.number_of_edges())
#####################################################################
sFileName=Base + '/' + Company + '/' + sOutputFileName1
print('###################################')
print('Storing :',sFileName)
print('###################################')
nx.write_gml(G, sFileName)
#####################################################################
sFileName=Base + '/' + Company + '/' + sOutputFileName2
print('###################################')
print('Storing Graph Image:',sFileName)
print('###################################')
plt.figure(figsize=(15, 15))
pos=nx.spectral_layout(G,dim=2)
nx.draw_networkx_nodes(G,pos, node_color='k', node_size=10, alpha=0.8)
nx.draw_networkx_edges(G, pos,edge_color='r', arrows=False, style='dashed')
nx.draw_networkx_labels(G,pos,font_size=12,font_family='sans-serif',font_color='b')
plt.axis('off')
plt.savefig(sFileName,dpi=600)
plt.show()
#####################################################################
print('###################################')
print('### Done!! ####################')
```

```
print('####################################')
##############################################################################
```

Output:

```
In [63]: runfile('C:/VKHCG/01-Vermeulen/05-Organise/Organise-Network-Routing-Company.py', wdir='C:/VKHCG/01-
Vermeulen/05-Organise')

################################
Working Base : C:/VKHCG  using  win32
################################
################################
Loading : C:/VKHCG/01-Vermeulen/02-Assess/01-EDS/02-Python/Assess-Network-Routing-Company.csv
################################
################################
  Company_Country_Code  ...      Company_Country_Name
0                   US  ...  United States of America
1                   US  ...  United States of America
2                   US  ...  United States of America
3                   US  ...  United States of America
4                   US  ...  United States of America

[5 rows x 5 columns]
(150, 5)
Nodes: 6
Edges: 6
################################
Storing : C:/VKHCG/01-Vermeulen/05-Organise/01-EDS/02-Python/Organise-Network-Routing-Company.gml
################################
################################
Storing Graph Image: C:/VKHCG/01-Vermeulen/05-Organise/01-EDS/02-Python/Organise-Network-Routing-Company.png
################################
```

Practical 9B
Directed Acyclic Graph

```
#######################################################################
import networkx as nx
import matplotlib.pyplot as plt
import sys
import os
import pandas as pd
#######################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + 'VKHCG'
else:
   Base='C:/VKHCG'
#######################################################################
print('#################################')
print('Working Base :',Base, ' using ', sys.platform)
print('#################################')
#######################################################################
sInputFileName='01-Retrieve/01-EDS/02-Python/Retrieve_Router_Location.csv'
sOutputFileName1='Assess-DAG-Company-Country.png'
sOutputFileName2='Assess-DAG-Company-Country-Place.png'
Company='01-Vermeulen'
#######################################################################
### Import Company Data
#######################################################################
sFileName=Base + '/' + Company + '/' + sInputFileName
print('#################################')
print('Loading :',sFileName)
print('#################################')
CompanyData=pd.read_csv(sFileName,header=0,low_memory=False, encoding="latin-1")
print('Loaded Company :',CompanyData.columns.values)
print('#################################')
#######################################################################
print(CompanyData)
print('#################################')
print('Rows : ',CompanyData.shape[0])
print('#################################')
#######################################################################
G1=nx.DiGraph()
G2=nx.DiGraph()
```

```
#####################################################################
for i in range(CompanyData.shape[0]):
   G1.add_node(CompanyData['Country'][i])
   sPlaceName= CompanyData['Place_Name'][i] + '-' + CompanyData['Country'][i]
   G2.add_node(sPlaceName)

print('###################################')
for n1 in G1.nodes():
   for n2 in G1.nodes():
      if n1 != n2:
         print('Link :',n1,' to ', n2)
         G1.add_edge(n1,n2)
print('###################################')
print("Nodes of graph: ")
print(G1.nodes())
print("Edges of graph: ")
print(G1.edges())
print('###################################')
#####################################################################
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
   os.makedirs(sFileDir)
#####################################################################
sFileName=sFileDir + '/' + sOutputFileName1
print('###################################')
print('Storing :', sFileName)
print('###################################')
nx.draw(G1,pos=nx.spectral_layout(G1),
      nodecolor='r',edge_color='g',
      with_labels=True,node_size=8000,
      font_size=12)
plt.savefig(sFileName) # save as png
plt.show() # display
#####################################################################
print('###################################')
for n1 in G2.nodes():
   for n2 in G2.nodes():
      if n1 != n2:
         print('Link :',n1,' to ', n2)
         G2.add_edge(n1,n2)
```

```
print('###################################')
print("Nodes of graph: ")
print(G2.nodes())
print("Edges of graph: ")
print(G2.edges())
print('###################################')
sFileDir=Base + '/' + Company + '/02-Assess/01-EDS/02-Python'
if not os.path.exists(sFileDir):
   os.makedirs(sFileDir)
#########################################################################
sFileName=sFileDir + '/' + sOutputFileName2
print('###################################')
print('Storing :', sFileName)
print('###################################')
nx.draw(G2,pos=nx.spectral_layout(G2),
      nodecolor='r',edge_color='b',
      with_labels=True,node_size=8000,
      font_size=12)
plt.savefig(sFileName) # save as png
plt.show() # display
```

Output:

```
In [16]: runfile('C:/VKHCG/01-Vermeulen/02-Assess/Assess-DAG-
Location.py', wdir='C:/VKHCG/01-Vermeulen/02-Assess')
################################
Working Base : C:/VKHCG  using  win32
################################
################################
Loading : C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/
Retrieve_Router_Location.csv
################################
Loaded Company : ['Country' 'Place_Name' 'Latitude' 'Longitude']
################################
    Country Place_Name  Latitude  Longitude
0        US   New York   40.7528   -73.9725
1        US   New York   40.7214   -74.0052
2        US   New York   40.7662   -73.9862
3        US   New York   40.7449   -73.9782
4        US   New York   40.7605   -73.9933
5        US   New York   40.7588   -73.9680
6        US   New York   40.7637   -73.9727
7        US   New York   40.7553   -73.9924
8        US   New York   40.7308   -73.9975
9        US   New York   40.7694   -73.9609
10       US   New York   40.7330   -74.0078
11       US   New York   40.7505   -73.9931
12       US   New York   40.7517   -73.9972
13       US   New York   40.7082   -74.0132
14       US   New York   40.7267   -73.9981
15       US   New York   40.7168   -73.9861
16       US   New York   40.7317   -73.9885
17       US   New York   40.7584   -73.9794
18       US   New York   40.7592   -73.9778
19       US   New York   40.7055   -74.0050
20       US   New York   40.6888   -74.0203
21       US   New York   40.7089   -74.0012
22       US   New York   40.7391   -73.9826
```

Practical 9 C
Graphics

CODE

```
import sys
import os
import pandas as pd
import matplotlib as ml
from matplotlib import pyplot as plt
#####################################################################
if sys.platform == 'linux':
   Base=os.path.expanduser('~') + '/VKHCG'
else:
   Base='C:/VKHCG'
print('##################################')
print('Working Base :',Base, ' using ', sys.platform)
print('##################################')
#####################################################################
GBase = Base+'/01-Vermeulen/06-Report/01-EDS/02-Python/'
ml.style.use('ggplot')

data=[
['London',          29.2, 17.4],
['Glasgow',         18.8, 11.3],
['Cape Town',    15.3, 9.0],
['Houston',         22.0, 7.8],
['Perth', 18.0, 23.7],
['San Francisco',             11.4, 33.3]
]
os_new=pd.DataFrame(data)
pd.Index(['Item', 'Value', 'Value Percent', 'Conversions', 'Conversion Percent',
      'URL', 'Stats URL'],
     dtype='object')

os_new.rename(columns = {0 : "Warehouse Location"}, inplace=True)
os_new.rename(columns = {1 : "Profit 2016"}, inplace=True)
os_new.rename(columns = {2 : "Profit 2017"}, inplace=True)

explode = (0, 0, 0, 0, 0, 0.1)
labels=os_new['Warehouse Location']
```

Practical 10: Data Visualization with Power BI.

A. Importing Data from excel

Step 1: Connect to an excel workbook.

1. Launch Power BI Desktop
2. From the Home ribbon, select Get Data. Excel is one of the Most Common data connections, so you can select if directly from the Get Data menu.
3. If you select the Get Data button directly, you can also select File>Excel and select connect.
4. In the Open File dialog box, select the Product.xlsx file (You need to download this file from http://services.odata.org/V3/Northwind/Northwind.svc/).

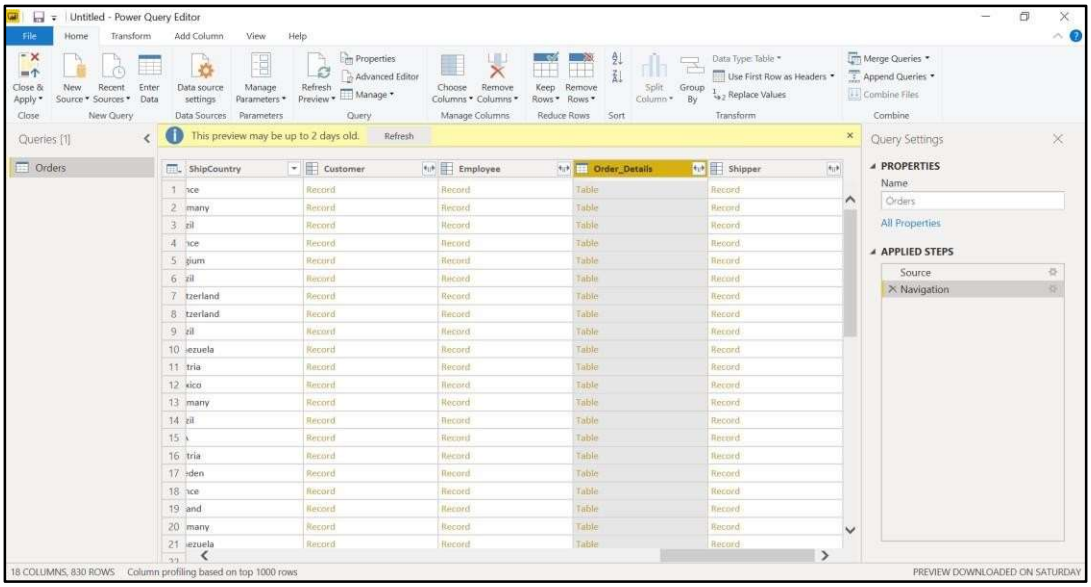
Step 2: We need to remove other columns except ProductID, ProductName, QuantityPerUnit and UnitInStock

Step 3: Change the data type of the UnitsInStock column

For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the UnitsInStock column's datatype is Whole Number.

1. Select the UnitsInStock column.
2. Select the Data Type drop-down button in the Home ribbon.
3. If not already a Whole Number, select Whole Number for data type from the drop down (Data Type: button also displays the data type for the current selection).

Output:



B. Importing data from Odata feed

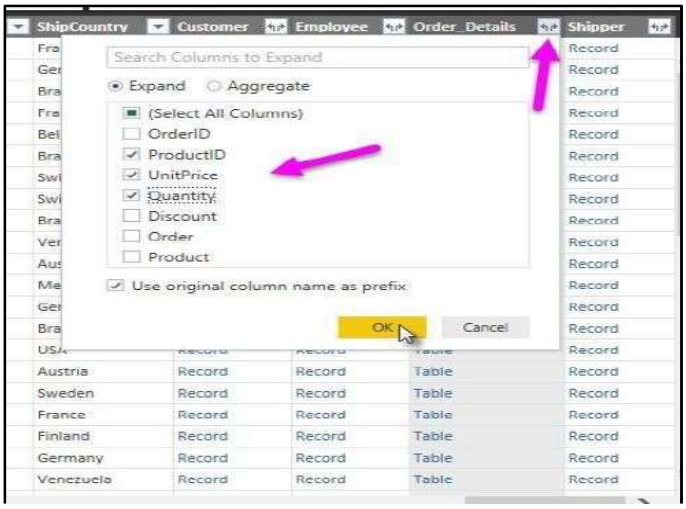
Step 1: Connect to an OData feed

1. From the Home ribbon tab in Query Editor, select Get Data.
2. Browse to the OData Feed data source.
3. In the OData Feed dialog box, paste the URL for the Northwind OData feed.
4. Select OK.
5. Now we need to select the order table from list of table. And then click on Transform Data.

Step 2: Expand the Order_Details table.

	Customer	Employee	Order_Details.ProductID	Order_Details.UnitPrice	Order_Details.Quantity	Shipper
1	Record	Record	11	14	12	Record
2	Record	Record	42	9.8	10	Record
3	Record	Record	72	34.8	5	Record
4	Record	Record	14	18.6	9	Record
5	Record	Record	51	42.4	40	Record
6	Record	Record	41	7.7	10	Record
7	Record	Record	51	42.4	35	Record
8	Record	Record	65	16.8	15	Record
9	Record	Record	22	16.8	6	Record
10	Record	Record	57	15.6	15	Record
11	Record	Record	65	16.8	20	Record
12	Record	Record	20	64.8	40	Record
13	Record	Record	33	2	25	Record
14	Record	Record	60	27.2	40	Record
15	Record	Record	31	10	20	Record
16	Record	Record	39	14.4	42	Record
17	Record	Record	49	16	40	Record
18	Record	Record	24	3.6	15	Record
19	Record	Record	55	19.2	21	Record
20	Record	Record	74	8	21	Record
21	Record	Record	2	15.2	20	Record

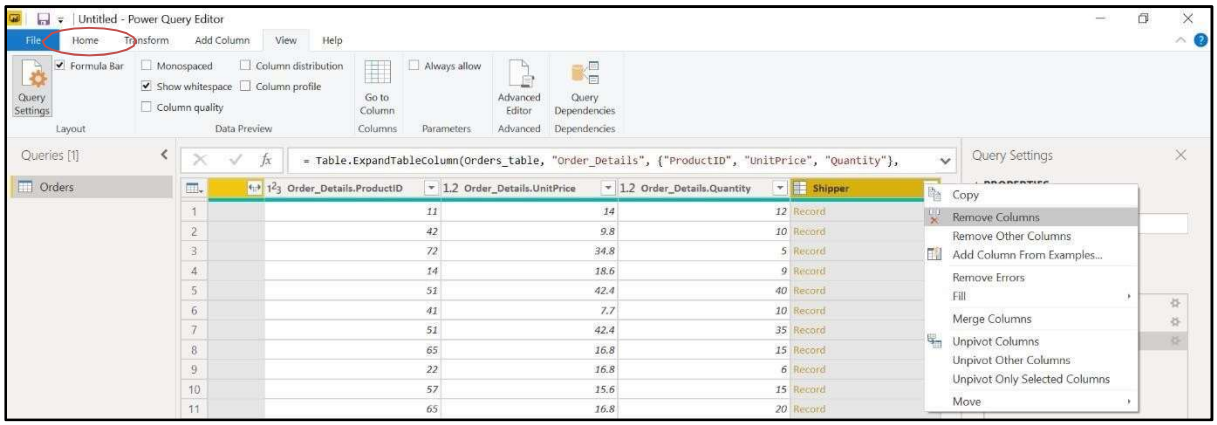
1. As per above picture drag the bottom scroll to the end and you will find the Order_Details column containing the tables.
2. Click on the expand symbol and then mark the checkbox as shown below



Step 3: Remove other columns to only display columns of interest

Remove all columns except OrderDate, ShipCity, ShipCountry, Order_Details.ProductID, Order_Details.UnitPrice, and Order_Details.Quantity columns.

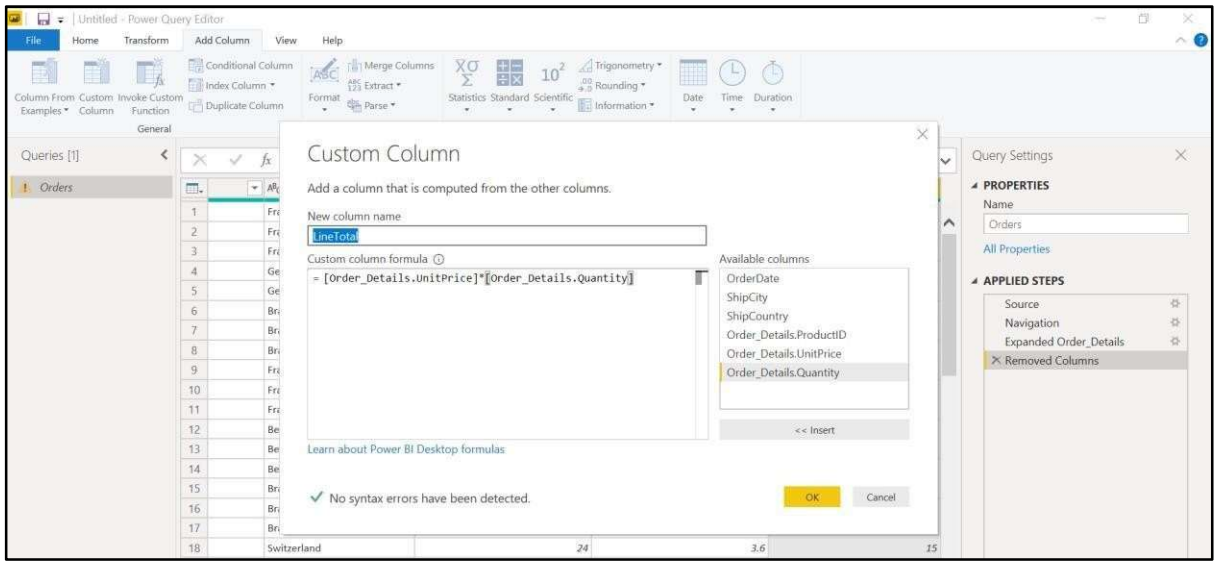
- Check mark the Formula Bar and then perform the below shown steps



Step 4: Calculate the line total for each Order_Details row

Calculate the line total for each Order_Details row:

1. In the Add Column ribbon tab, click Add Custom Column.
2. In the Add Custom Column dialog box, in the Custom Column Formula textbox, enter [Order_Details.UnitPrice] * [Order_Details.Quantity].
3. In the New column name textbox, enter LineTotal.



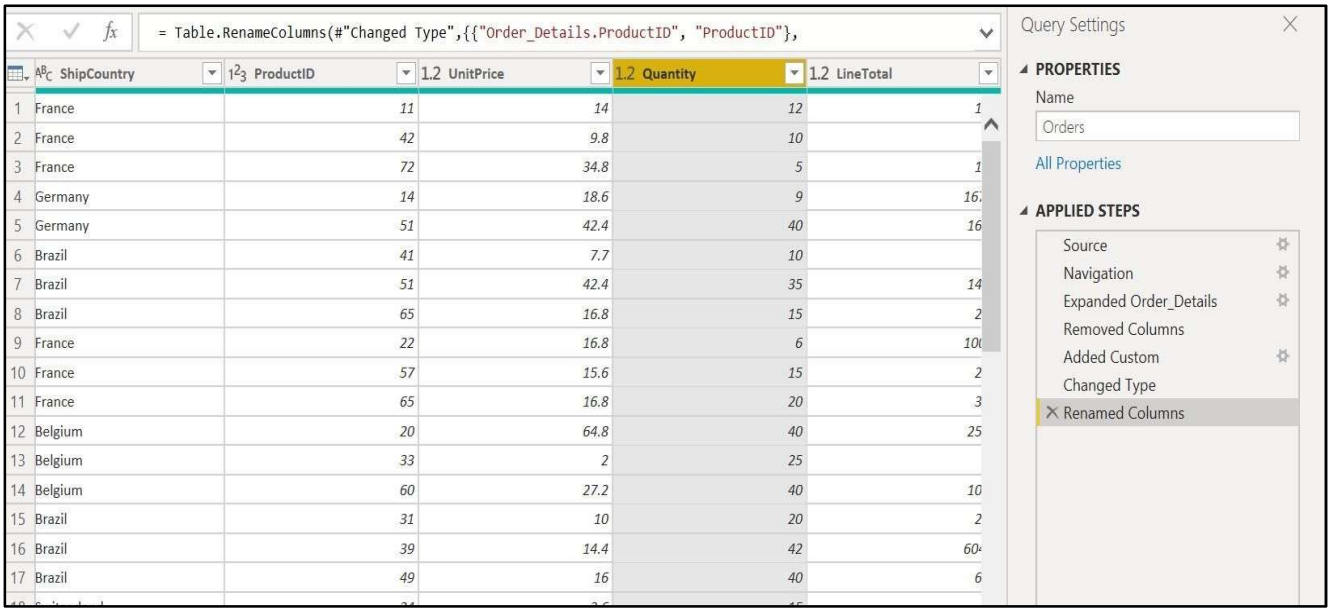
Step 5: Set the datatype of the LineTotal field

1. Right click the LineTotal column.
2. Select Change Type and choose Decimal Number.

Step 6: Rename and reorder columns in the query

1. In Query Editor, drag the LineTotal column to the left, after ShipCountry.

2. Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.



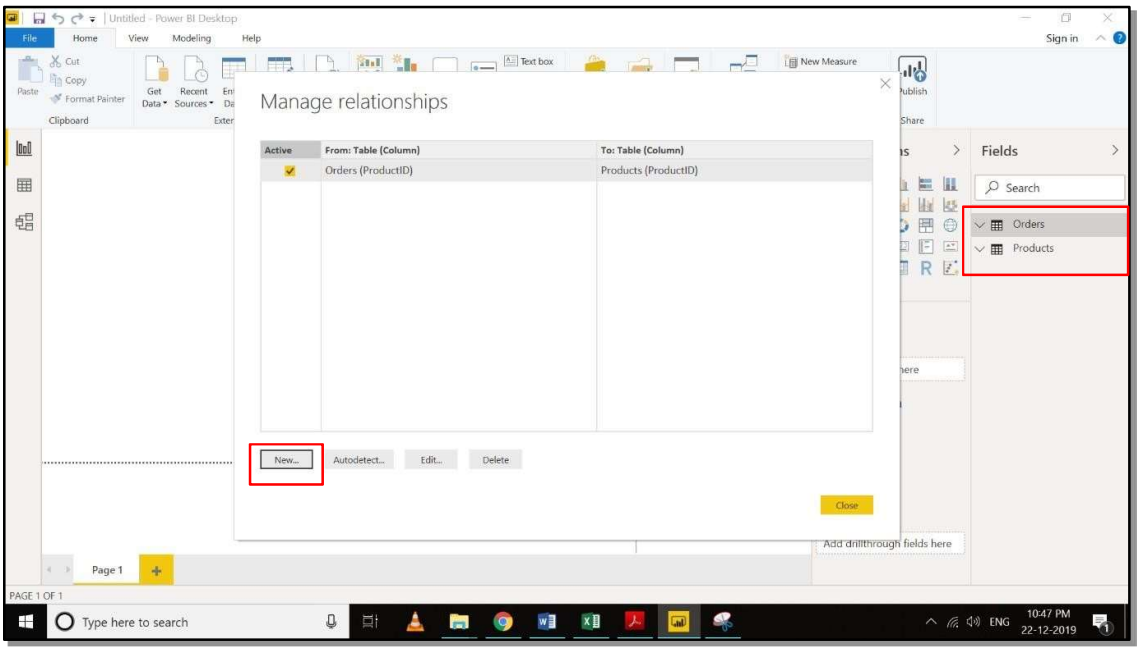
	ShipCountry	ProductID	UnitPrice	Quantity	LineTotal
1	France	11	14	12	1
2	France	42	9.8	10	
3	France	72	34.8	5	1
4	Germany	14	18.6	9	16.
5	Germany	51	42.4	40	16
6	Brazil	41	7.7	10	
7	Brazil	51	42.4	35	14
8	Brazil	65	16.8	15	2
9	France	22	16.8	6	10
10	France	57	15.6	15	2
11	France	65	16.8	20	3
12	Belgium	20	64.8	40	25
13	Belgium	33	2	25	
14	Belgium	60	27.2	40	10
15	Brazil	31	10	20	2
16	Brazil	39	14.4	42	60
17	Brazil	49	16	40	6

	ShipCountry	Order_Details.ProductID
1	France	11
2	France	42
3	France	72
4	Germany	14
5	Germany	51
6	Brazil	41
7	Brazil	51
8	Brazil	65
9	France	22
10	France	57
11	France	65
12	Belgium	20
13	Belgium	33
14	Belgium	60
15	Brazil	31
16	Brazil	39
17	Brazil	49
18	Switzerland	24
19	Switzerland	55

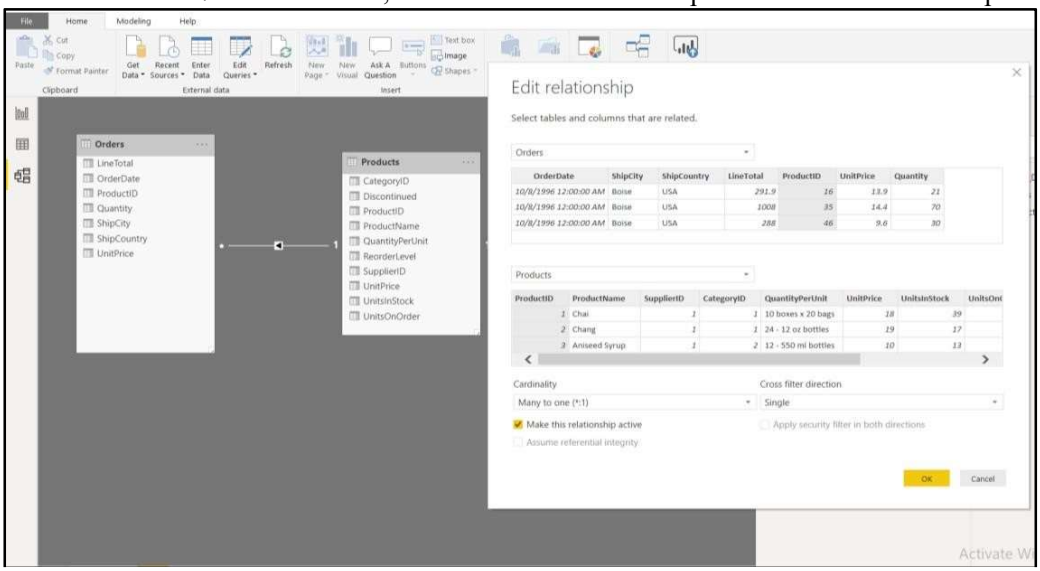
C. Data visualization with Power BI.

For this we need to first load the Products.xlsx table in the Power BI.

1. Go to Get Data and click
2. Select excel and click
3. Now select your Product.xlsx workbook
4. Click Open
5. Now click on Manage Relationships in Home ribbon



6. When we attempt to create the relationship, we see that one already exists! As shown in the Create Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.
7. Select Cancel, and then select Relationship view in Power BI Desktop.



Smt . Durgadevi Sharma Charitable Trust's
Chandrabhan Sharma College
of Arts, Commerce & Science
(Autonomous)
Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

Subject Name : **Security Breaches and Counter Measures**

Name : **PRATIK JAGDISHCHANDRA MAURYA**

Seat No : **1310226**

Teaching Faculty : **Mr. Nitesh Shukla**

DEPARTMENT OF INFORMATION TECHNOLOGY

CHANDRABHAN SHARMA COLLEGE OF ARTS,

COMMERCE & SCIENCE

NAAC RE-ACCREDITED 'A' Grade (CGPA 3.10)

(Autonomous)

Smt . Durgadevi Sharma Charitable Trust's
Chandrabhan Sharma College
of Arts, Commerce & Science
(Autonomous)
Hindi Linguistic Minority Institution
(Affiliated to University of Mumbai)
NAAC RE-ACCREDITED 'A' GRADE (CGPA 3.10)
Adi Shankaracharya Marg, Powai Vihar Complex, Powai, Mumbai - 400 076

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that Mr./~~Miss~~ **PRATIK JAGDISHCHANDRA MAURYA** having Exam SeatNo. **1310226** of M.Sc.IT **(Semester I)** has complete the Practical work in the subject of "**Security Breaches and Counter Measures**" during the **Academic year 2024-25** under the guidance of **Mr. Nitesh Shukla** being the Partial requirement for The fulfilment of the curriculum of Degree of Master of Science in Information Technology, University of Mumbai.

Internal Examiner **Co-Ordinator**

Date: **College Seal**

External Examiner

Dates	Topic	Sign
	Practical No. 1	
	A. Recon-ng	
	B. Windows Command Line Utilities a. Ping b. Tracert c. Tracert using Ping d. NSLookup	
	C. HTTrack	
	D. Metasploit	
	E. DNS WhoIsLookup	
	F. Smart WhoIs	
	G. eMailTracker Pro	
	Practical No. 2	
	A. Hping2 for DoS attack (Kali Linux)	
	B. Advanced IP Scanner	
	C. Angry IP Scanner	
	D. Masscan (Kali Linux)	
	E. Scanning open ports of the system using CurrPorts	
	F. Create a TCP, UDP or SNMP packet using Colasoft Packet Builder	
	G. TheDude	
	Practical No. 3	
	A. Use Proxy Workbench Tool	
	B. Perform Network Discovery using following tools: a LANState Pro b Network View c OpManager	
	Practical No. 4	

	A. Perform Enumeration using the following tools: a Nmap b NetBIOS c Hyena d SuperScan Software e Wireshark	
	B. Perform Vulnerability Analysis using Nessus.	
	Practical No. 5	
	A. Winrtgen	
	B. PWDump	
	C. Ophcrack	
	D. NTFS Stream Manipulation	
	E. ADS Spy	
	F. Quickstego	
	Practical No. 6	
	A. Sniff the network packet to break the password using Wireshark.	
	B. Change the MAC of the system using SMAC tool.	
	C. Perform the network analysis using Caspa Network Analyzer Tool.	
	Practical No. 7	
	A. Use the social engineering toolkit to perform social engineering attack.	
	B. Perform the DDoS Attack on a website using: a Golden Key b Metasploit c HOIC LOIC	
	Practical No. 8	
	A. Perform the Web Scanning using OWSAP Zed Proxy.	
	B. Use the HoneyBOT to capture malicious network traffic.	
	Practical No. 9	
	A. Protect the web application using dotDefender.	

	B. Perform the database attack using SQL Injection Technique.	
	Practical No. 10	
	A. HashCalc	
	B. CrypTool	
	C. TrueCrypt	

Practical No. 1

Aim: Perform the footprinting and reconnaissance using the tools.

A. **Recon-ng**
B. **Windows Command Line Utilities**
   a. **Ping**
   b. **Tracert**
   c. **Tracert using Ping**
   d. **NSLookup**
C. **HTTrack**
D. **Metasploit**
E. **DNS WhoIsLookup**
F. **Smart Whois**
G. **eMailTracker Pro**

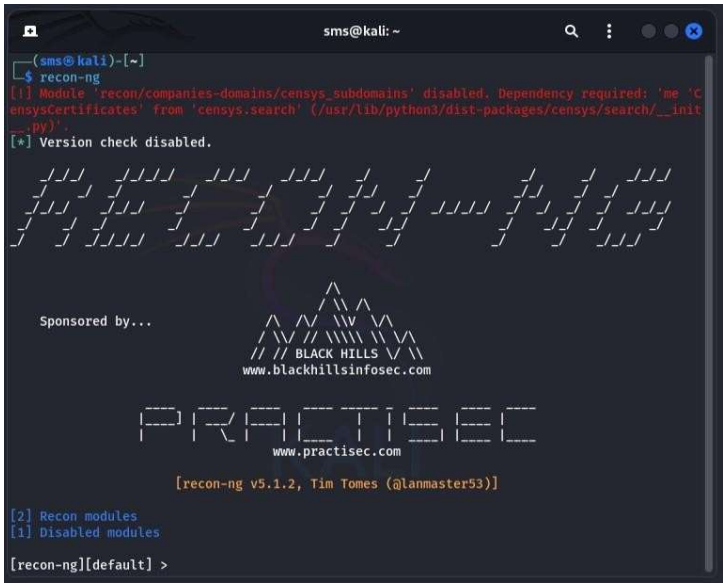
Footprinting and Reconnaissance

Footprinting and reconnaissance are used to collect basic information about the target systems in order to exploit them. The target information is IP location information, routing information, business information, address, phone number and DNS records.

A. Recon-ng

Recong-ng is a full feature Web Reconnaissance framework used for information gathering purpose as well as network detection. This tool is written in python, having independent modules, database interaction and other features. You can download the software from www.bitbucket.org. This Open Source Web Reconnaissance tool requires Kali Linux Operating system.

1. Open the terminal of Kali Linux and type the command ***recon-ng***.



2. Create a new workspace using command ***workspaces create <workspace>***.

```
[recon-ng][default] > workspaces create CEH
```

3. Add the target domain to perform a network recon using command ***db insert domains***.

```
[recon-ng][CEH] > db insert domains
domain (TEXT): certifiedhacker.com
notes (TEXT): Hacking Website
[*] 1 rows affected.
```

4. View the added domain by typing ***show domains***.

```
[recon-ng][CEH] > show domains

  +-------------------------------------------------------------+
  | rowid |       domain       |      notes      |    module    |
  +-------------------------------------------------------------+
  | 1     | certifiedhacker.com | Hacking Website | user_defined |
  +-------------------------------------------------------------+

[*] 1 rows returned
```

Recon-ng works with independent modules, database interaction, built in convenience functions, interactive help, and command completion, Recon-ng provides a powerful environment in which open source web-based reconnaissance can be conducted quickly and thoroughly. To add new modules you will use marketplace.

5. View the entire marketplace using command ***marketplace search***.

```
[recon-ng][CEH] > marketplace search

  +-------------------------------------------------------------------------------------------+
  |                          Path                          | Version |     Status    | Updated    | D | K |
  +-------------------------------------------------------------------------------------------+
| recon/domains-hosts/google_site_web                      | 1.0     | installed     | 2019-06-24 |   |   |
| recon/domains-hosts/hackertarget                         | 1.1     | installed     | 2020-05-17 |   |   |
```

6. Install required recon-ng using command ***marketplace install <module>***.

```
[recon-ng][default] > marketplace install recon/domains-hosts/hackertarget
```

7. View the installed modules by typing ***modules search***.

```
[recon-ng][CEH] > modules search

  Recon
  -----
    recon/domains-hosts/google_site_web
    recon/domains-hosts/hackertarget
```

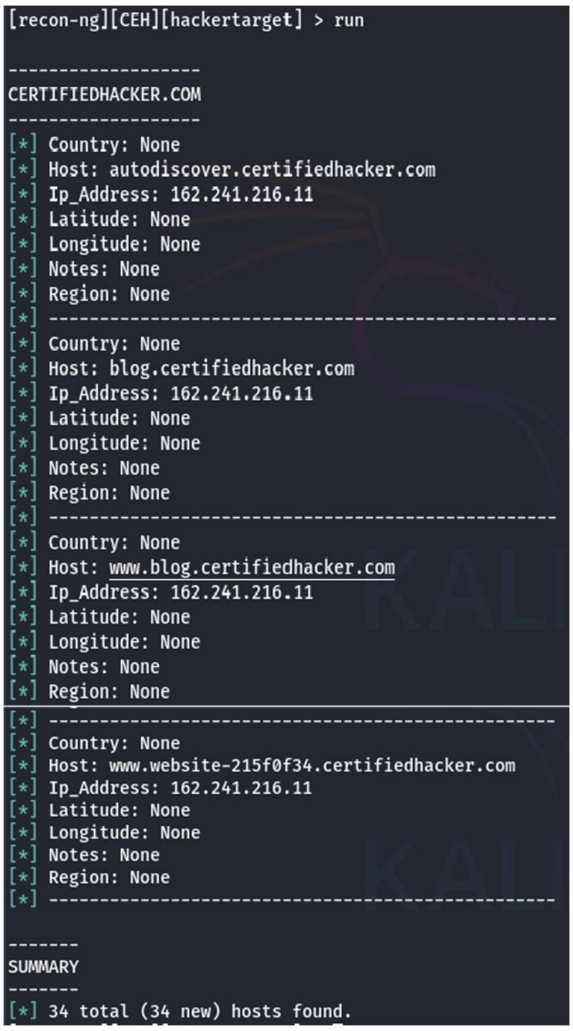
8. Load a specific module using command ***modules load <module>***.

```
[recon-ng][CEH] > modules load recon/domains-hosts/hackertarget
```

HackerTarget provides various services to gather information about domains, IP addresses, and other entities.

9. View information about the particular module by typing ***info***.

10. Run the module by typing ***run***.



11. Change the source using command ***options set SOURCE <domain>*** and view the target by typing ***input***.

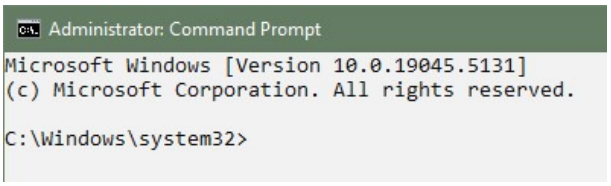
B. Windows Command Line Utilities

Windows Command Line Utilities are tools that allow users to interact with the Windows operating system using text-based commands. They provide access to system functions without a graphical interface, enabling tasks like file management, system diagnostics, and network troubleshooting.

a. Ping

The ping command sends ICMP (Internet Control Message Protocol) Used to test the reachability of a host on a IP network and measures the travel time for messages sent from the originating host to destinantion target.

1. Open Windows Command Line (cmd) from Windows PC.



2. Enter the command ***ping <domain name>***. (eg. www.certifiedhacker.com)

```
C:\Windows\system32>ping www.certifiedhacker.com

Pinging certifiedhacker.com [162.241.216.11] with 32 bytes of data:
Reply from 162.241.216.11: bytes=32 time=243ms TTL=45
Reply from 162.241.216.11: bytes=32 time=247ms TTL=45
Reply from 162.241.216.11: bytes=32 time=240ms TTL=45
Reply from 162.241.216.11: bytes=32 time=241ms TTL=45

Ping statistics for 162.241.216.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 240ms, Maximum = 247ms, Average = 242ms
```

From the output, you can observe and extract the following information:

i. certifiedhacker.com is live
ii. IP Address of certifiedhacker.com
iii. Round Trip Time
iv. TTL value
v. Packet Loss Statistics

3. Use the last command and add the ***-f*** parameter to not fragment on the ping packet and
   -l to set the frame size to ***1500*** bytes.

ping <domain name> -f -l <frame size>

```
C:\Windows\system32>ping www.certifiedhacker.com -f -l 1500

Pinging certifiedhacker.com [162.241.216.11] with 1500 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 162.241.216.11:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This message above means that the frame is too large to be on the network and needs to be fragmented.

```
C:\Windows\system32>ping www.certifiedhacker.com -f -l 1473

Pinging certifiedhacker.com [162.241.216.11] with 1473 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 162.241.216.11:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
C:\Windows\system32>ping www.certifiedhacker.com -f -l 1472

Pinging certifiedhacker.com [162.241.216.11] with 1472 bytes of data:
Reply from 162.241.216.11: bytes=1472 time=247ms TTL=45
Reply from 162.241.216.11: bytes=1472 time=243ms TTL=45
Reply from 162.241.216.11: bytes=1472 time=245ms TTL=45
Reply from 162.241.216.11: bytes=1472 time=244ms TTL=45

Ping statistics for 162.241.216.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 243ms, Maximum = 247ms, Average = 244ms
```

The propose here is to try different values until you reach the maximum frame size. In conclusion, 1472 bytes shows the maximum frame size on this machine's network.

b. Tracert

Tracert (short for "trace route") is a command-line network diagnostic tool used to track the path data takes from one device to another across an IP network. It identifies each hop (router) on the path and measures the delay (latency) for each one. This can help diagnose network issues or understand the route data takes over the internet or a local network.

1. Open a new window on your prompt or powershell and type:

tracert <domain name>

```
C:\Windows\system32>tracert www.certifiedhacker.com

Tracing route to certifiedhacker.com [162.241.216.11]
over a maximum of 30 hops:

  1     6 ms     3 ms     4 ms  reliance.reliance [192.168.29.1]
  2     6 ms     7 ms     6 ms  10.227.200.1
  3     5 ms     5 ms     5 ms  172.31.2.26
  4     9 ms    12 ms     8 ms  192.168.53.186
  5     8 ms     8 ms     8 ms  172.26.76.214
  6     9 ms     7 ms     8 ms  172.26.76.194
  7    10 ms     9 ms     6 ms  192.168.53.174
  8     *        *        *     Request timed out.
  9     *        *        *     Request timed out.
 10    14 ms    11 ms    11 ms  103.198.140.176
 11   109 ms   106 ms   109 ms  103.198.140.54
 12     *        *        *     Request timed out.
 13   111 ms   127 ms   110 ms  mei-b5-link.ip.twelve99.net [62.115.11.140]
 14     *        *        *     Request timed out.
 15   128 ms   124 ms   125 ms  ldn-bb1-link.ip.twelve99.net [62.115.135.24]
 16     *        *        *     Request timed out.
 17   263 ms     *        *     chi-bb1-link.ip.twelve99.net [62.115.139.33]
 18   242 ms   244 ms   240 ms  den-bb1-link.ip.twelve99.net [62.115.115.76]
 19   296 ms   304 ms   305 ms  den-bb2-link.ip.twelve99.net [62.115.140.89]
 20   244 ms   245 ms   245 ms  salt-b4-link.ip.twelve99.net [62.115.132.207]
 21   245 ms   244 ms   246 ms  salt-b5-link.ip.twelve99.net [62.115.136.107]
 22   259 ms   258 ms   259 ms  newfolddigital-ic-380138.ip.twelve99-cust.net [80.239.167.103]
 23   258 ms   259 ms   259 ms  69-195-64-105.unifiedlayer.com [69.195.64.105]
 24   256 ms   253 ms   255 ms  po99.prv-leaf1b.net.unifiedlayer.com [162.144.240.135]
 25   261 ms   269 ms   264 ms  box5331.bluehost.com [162.241.216.11]

Trace complete.
```

The system resolves the URL into its IP address and starts to trace the path to the destination. Here it takes 25 hops for the packet to reach the specified destination.

2. Show different options for the command: ***tracert /?***

```
C:\Windows\system32>tracert /?

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
```

c. Tracert using Ping

If tracert is unavailable, you can use ping in an iterative way. Every frame on the network has their own TTL defined. If the TTL reaches 0, the router discards the packet to prevent packet loss. Use this feature to gradually increase the TTL and find each hop along the path.

1. Open a new window on your prompt or powershell and type:

ping <domain name> -i <hop count>

-i parameter specifies the Time To Live (TTL), which controls how many hops a packet can take before it's discarded. (values between ***1-255***).

```
C:\Windows\system32>ping www.certifiedhacker.com -i 3 -n 1

Pinging certifiedhacker.com [162.241.216.11] with 32 bytes of data:
Reply from 172.31.2.26: TTL expired in transit.

Ping statistics for 162.241.216.11:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
```

```
C:\Windows\system32>ping www.certifiedhacker.com -i 24 -n 1

Pinging certifiedhacker.com [162.241.216.11] with 32 bytes of data:
Reply from 162.144.240.135: TTL expired in transit.

Ping statistics for 162.241.216.11:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
```

```
C:\Windows\system32>ping www.certifiedhacker.com -i 25 -n 1

Pinging certifiedhacker.com [162.241.216.11] with 32 bytes of data:
Reply from 162.241.216.11: bytes=32 time=258ms TTL=44

Ping statistics for 162.241.216.11:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 258ms, Maximum = 258ms, Average = 258ms
```

TTL expired means that the router discarded the frame, because the TTL has expired

(reached 0).

d. NSLookup

Nslookup (stands for "Name Server Lookup") is a useful command for getting information from the DNS server. It is used for querying the DNS (Domain Name System), to obtain a domain name or IP address mapping and other specific DNS record. It is also used to troubleshoot DNS-related problems.

1. Open a new window on your prompt or powershell and type: ***nslookup***
   This command will launch a interactive mode, you can type ***help*** to list available commands.

```
C:\Windows\system32>nslookup
Default Server:  reliance.reliance
Address:  2405:201:3e:f808::c0a8:1d01

> help
Commands:   (identifiers are shown in uppercase, [] means optional)
NAME            - print info about the host/domain NAME using default server
NAME1 NAME2     - as above, but use NAME2 as server
help or ?       - print info on common commands
set OPTION      - set an option
    all                 - print options, current server and host
    [no]debug           - print debugging information
    [no]d2              - print exhaustive debugging information
    [no]defname         - append domain name to each query
    [no]recurse         - ask for recursive answer to query
    [no]search          - use domain search list
    [no]vc              - always use a virtual circuit
    domain=NAME         - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
    root=NAME           - set root server to NAME
    retry=X             - set number of retries to X
    timeout=X           - set initial time-out interval to X seconds
    type=X              - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
    querytype=X         - same as type
    class=X             - set query class (ex. IN (Internet), ANY)
    [no]msxfr           - use MS fast zone transfer
    ixfrver=X           - current version to use in IXFR transfer request
server NAME     - set default server to NAME, using current default server
lserver NAME    - set default server to NAME, using initial server
root            - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a          -  list canonical names and aliases
    -d          -  list all records
    -t TYPE     -  list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE           - sort an 'ls' output file and view it with pg
exit            - exit the program
```

2. For query IP address of a given domain, you need to set the type to A record, then enter the target domain:

   > **type=a**

   > ***<domain name>***

```
> type=a
Server:  reliance.reliance
Address:  2405:201:3e:f808::c0a8:1d01

> www.certifiedhacker.com
Server:  reliance.reliance
Address:  2405:201:3e:f808::c0a8:1d01

Non-authoritative answer:
Name:    certifiedhacker.com
Address:  162.241.216.11
Aliases:  www.certifiedhacker.com
```

3. The Authoritative is a name server that has the original source files of a domain zone files. To obtain the Authoritative name server, set the ***type*** to ***CNAME*** record and query the target:

> **set type=cname**

> ***certifiedhacker.com***

```
> set type=cname
> certifiedhacker.com
Server:  reliance.reliance
Address:  2405:201:3e:f808::c0a8:1d01

certifiedhacker.com
        primary name server = ns1.bluehost.com
        responsible mail addr = dnsadmin.box5331.bluehost.com
        serial  = 2024111300
        refresh = 86400 (1 day)
        retry   = 7200 (2 hours)
        expire  = 3600000 (41 days 16 hours)
        default TTL = 300 (5 mins)
```

The ***CNAME*** lookup is done directly against the domain's authoritative name server.

4. With the authoritative name server, you can determine the IP address. To query IP address set the ***type*** to ***A***, then type the primary name server displayed in your lab environment, in my case: ***ns1.bluehost.com***.

> **set type=a**

> ***ns1.bluehost.com***

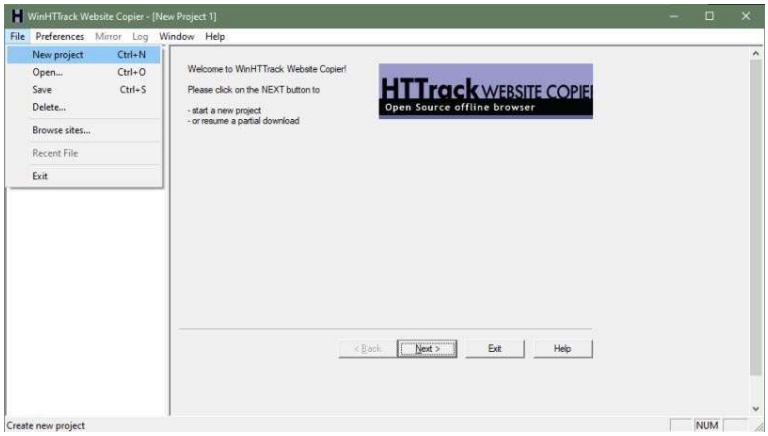
```
> set type=a
> ns1.bluehost.com
Server:  reliance.reliance
Address:  2405:201:3e:f808::c0a8:1d01

Non-authoritative answer:
Name:    ns1.bluehost.com
Address:  162.159.24.80
```

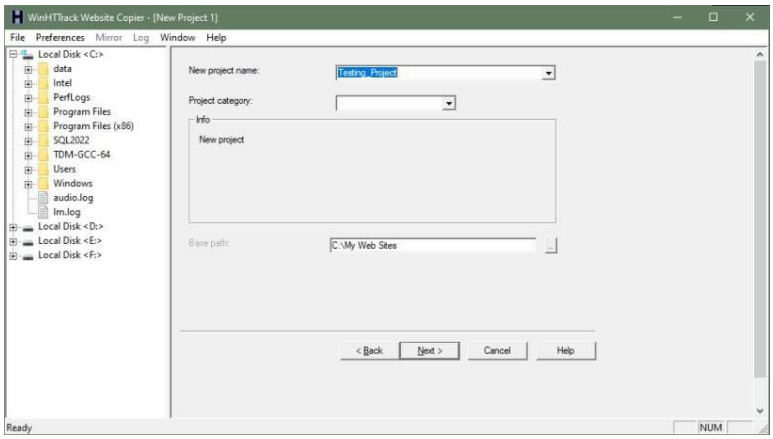
In conclusion, the Authoritative name server stores the records associated with the respective domain. Having the authoritative name server (primary name server) and the IP address associated with it, an attacker can attempt to exploit the server, performing attacks like DDoS, URL redirection and so on.

C. HTTrack

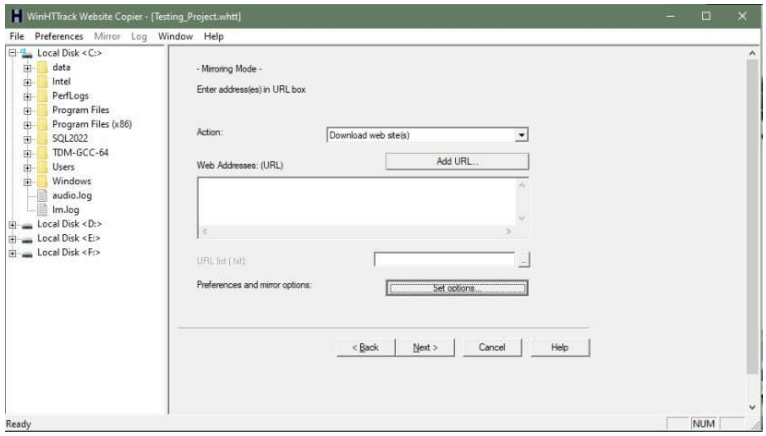
Web site copier is an offline browser utility that downloads a Web site to a local directory.
This application is available in GUI and CLI.
Works on Linux / OSX / BSD / Unix, Windows and Android.
Download and Install the WinHTTrack Website Copier Tool from the website: https://www.httrack.com/

1. Create a new project named '**Testing_Project**'.

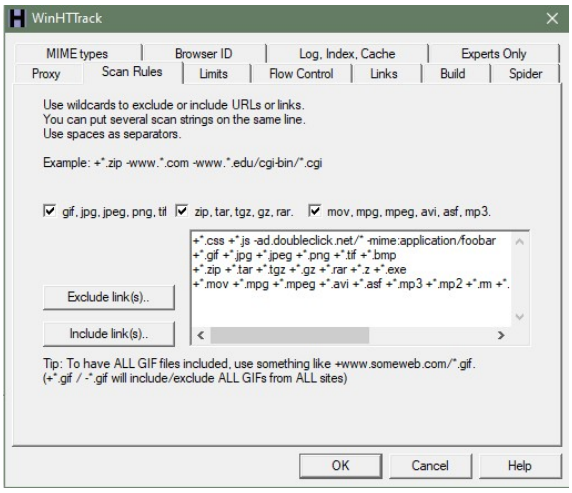




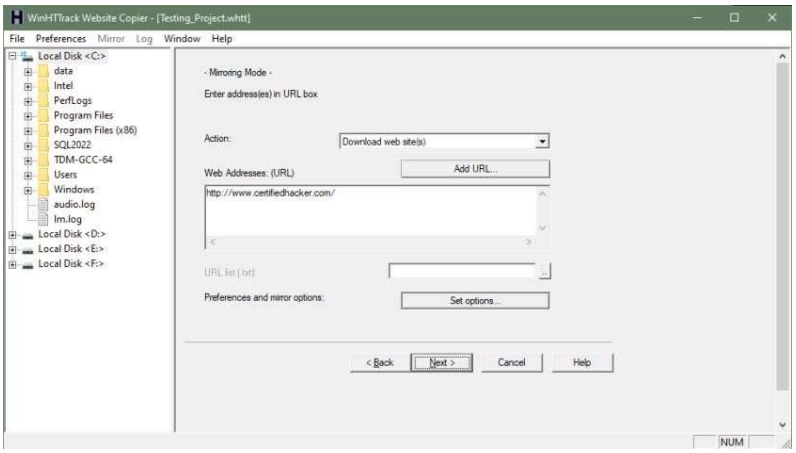
2. Click on **Set options…** button.



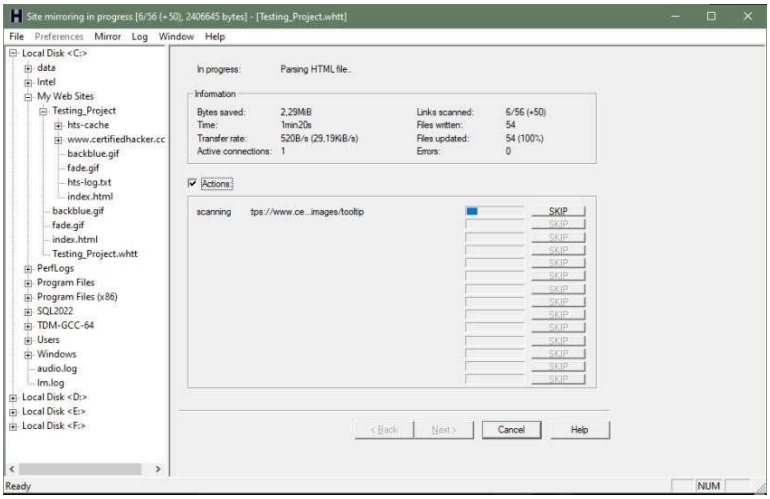
3. Go to **Scan Rules** tab and select options as required.



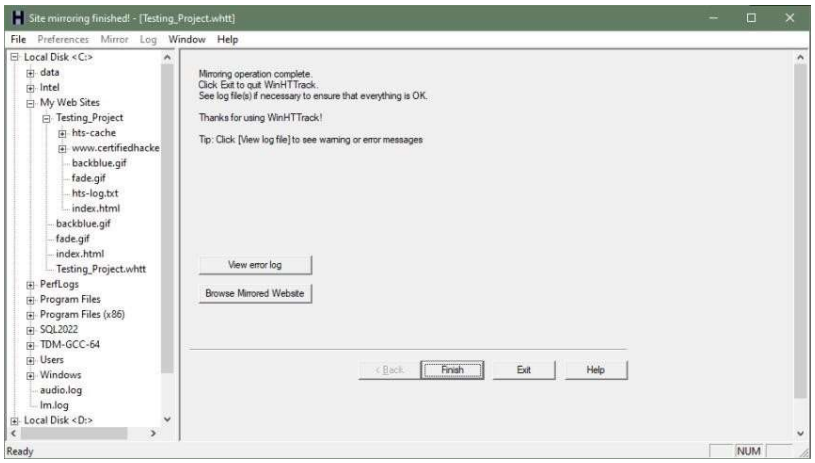
4. Enter the web address in the field and click **Next**.

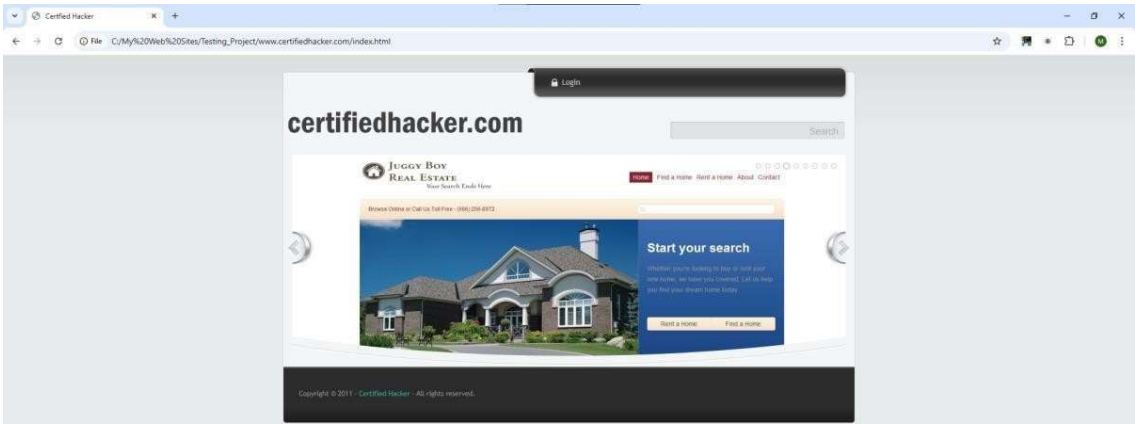


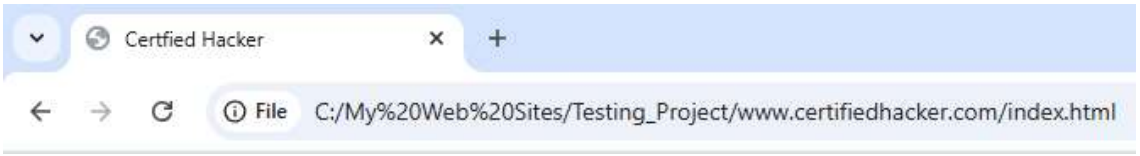
5. Click **Next**.



6. Click on **Browse Mirrored Website**.







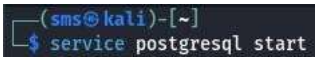
Observe the above website is copied into a local directory and browsed from there. Now you can explore the website in an offline environment for the structure of the website and other parameters. To make sure, compare the website to the original website.

D. Metasploit

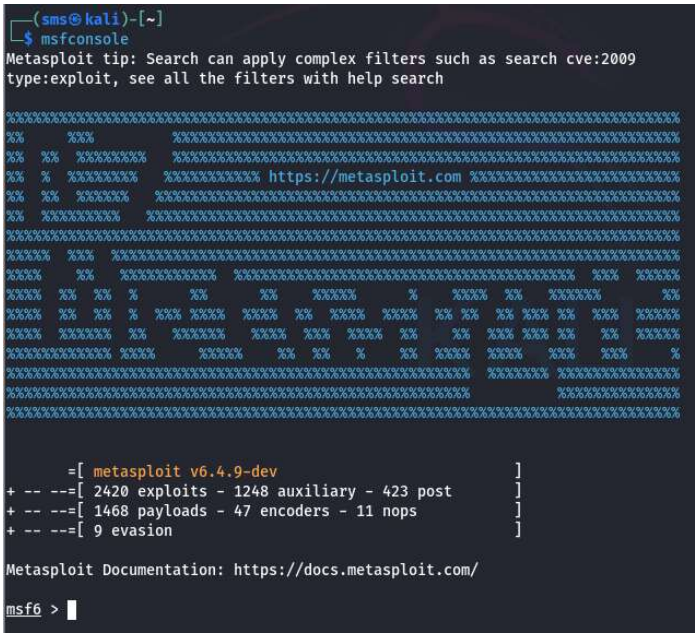
The Metasploit Framework is a tool that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. Metasploit can be used to test the vulnerability of computer systems or to break into remote systems. It can also be used to find number of alive hosts, scan for open ports and services, exploit vulnerabilities, pivot further into a network, collect evidence, and create a report of the test results.

1. Open a Terminal window. Start PostgreSQL database service to link with Metasploit:

service postgresql start

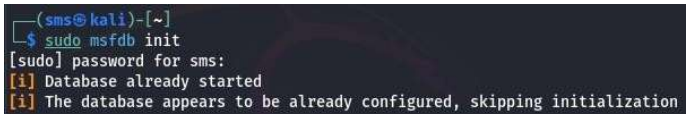


2. Now type ***msfconsole*** to launch Metasploit.

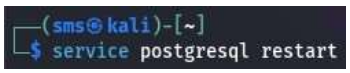


3. Check if Metasploit is connected to the database successfully: ***db_status***

If you got this message, it means that database did not connected to msf properly. To fix this issue, type ***exit*** to quit Metasploit.
Then, to initiate the database, type: ***msfdb init***



4. Then, restart the postgresql service: ***service postgresql restart***



5. Start Metasploit again and run the ***db_status*** to check the database status:

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
```

Now the database is connected successfully to the msf.

6. To scan the subnet, we can use Nmap: ***nmap -T5 -O -oX <file> <IP Range>***
Nmap starts scanning the subnet and showing the results on the screen.
The -T flag adjusts the timing template from T0 (slowest) to T5 (fastest).
The -oX Test Nmap command stands for output in XML file called Test.

```
msf6 > nmap -T5 -O -oX Test 162.241.216.0/8
[*] exec: nmap -T5 -O -oX Test 162.241.216.0/8

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 16:04 IST
```

7. We can import the Nmap results from the database: ***db_import Test***

```
msf6 > db_import Test
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 162.241.216.11
[*] Successfully imported /home/sms/Test
```

8. To see the hosts and their details discovered by Nmap type:

hosts

...
(Check the OS versions, IP and MAC addresses)

```
msf6 > hosts

Hosts
=====

address         mac  name                os_name   os_flavor  os_sp  purpose  info  comments
-------         ---  ----                -------   ---------  -----  -------  ----  --------
162.241.216.11       box5331.bluehost.com  embedded                    device
```

9. To scan to check the services running on this system, type the following command:

db_nmap -T4 -sS -A <domain name>

Nmap starts to footprint the system and list out the OS details.

```
msf6 > db_nmap -T4 -sS -A www.certifiedhacker.com
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 15:08 IST
[*] Nmap: Stats: 0:00:35 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
[*] Nmap: SYN Stealth Scan Timing: About 47.69% done; ETC: 15:10 (0:00:37 remaining)
[*] Nmap: Warning: 162.241.216.11 giving up on port because retransmission cap hit (6).
[*] Nmap: Stats: 0:02:23 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
[*] Nmap: Service scan Timing: About 93.33% done; ETC: 15:11 (0:00:01 remaining)
[*] Nmap: Stats: 0:03:25 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
[*] Nmap: NSE Timing: About 90.00% done; ETC: 15:12 (0:00:02 remaining)
[*] Nmap: Nmap scan report for www.certifiedhacker.com (162.241.216.11)
[*] Nmap: Host is up (0.050s latency).
[*] Nmap: rDNS record for 162.241.216.11: box5331.bluehost.com
[*] Nmap: Not shown: 942 closed tcp ports (reset), 43 filtered tcp ports (no-response)
[*] Nmap: PORT     STATE SERVICE    VERSION
[*] Nmap: 21/tcp   open  ftp        Pure-FTPd
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: | ssl-cert: Subject: commonName=*.bluehost.com
[*] Nmap: | Subject Alternative Name: DNS:*.bluehost.com, DNS:bluehost.com
[*] Nmap: | Not valid before: 2024-01-24T00:00:00
[*] Nmap: |_Not valid after:  2025-02-23T23:59:59
[*] Nmap: 22/tcp   open  ssh        OpenSSH 7.4 (protocol 2.0)
[*] Nmap: | ssh-hostkey:
[*] Nmap: |   1024 4a:2c:4a:5d:c6:46:51:63:1f:7f:52:69:51:04:9d:bd (DSA)
[*] Nmap: |_  2048 71:7b:91:d1:62:55:63:e6:20:40:48:03:e0:53:49:e0 (RSA)
[*] Nmap: 26/tcp   open  smtp       Exim smtpd 4.96.2
[*] Nmap: | smtp-commands: box5331.bluehost.com Hello www.certifiedhacker.com [49.43.24.20], SIZE 52428800, 8BITMIME, PIPELINING, PIPECONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP
[*] Nmap: |_ Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: 53/tcp   open  domain     ISC BIND 9.11.4-P2 (RedHat Enterprise Linux 7)
[*] Nmap: | dns-nsid:
[*] Nmap: |_  bind.version: 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.9
[*] Nmap: 80/tcp   open  http       Apache httpd
[*] Nmap: |_http-server-header: Apache
[*] Nmap: |_http-title: Did not follow redirect to https://www.certifiedhacker.com/
[*] Nmap: 110/tcp  open  pop3       Dovecot pop3d
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: |_pop3-capabilities: PIPELINING USER SASL(PLAIN LOGIN) RESP-CODES CAPA UIDL TOP STLS AUTH-RESP-CODE
[*] Nmap: 143/tcp  open  imap       Dovecot imapd
[*] Nmap: |_imap-capabilities: ENABLE IDLE SASL-IR listed AUTH=LOGINA0001 have AUTH=PLAIN LOGIN-REFERRALS more post-login ID capabilities LITERAL+ Pre-login NAMESPACE OK STARTTLS IMAP4rev1
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: 443/tcp  open  ssl/http   Apache httpd
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: | http-server-header:
[*] Nmap: |   Apache
[*] Nmap: |_  nginx/1.25.5
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: |_http-title: Certfied Hacker
[*] Nmap: 465/tcp  open  ssl/smtp   Exim smtpd 4.96.2
[*] Nmap: |_smtp-commands: Couldn't establish connection on port 465
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: 587/tcp  open  smtp       Exim smtpd 4.96.2
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: | smtp-commands: box5331.bluehost.com Hello www.certifiedhacker.com [49.43.24.20], SIZE 52428800, 8BITMIME, PIPELINING, PIPECONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP
[*] Nmap: |_ Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
[*] Nmap: 993/tcp  open  ssl/imap   Dovecot imapd
[*] Nmap: |_imap-capabilities: ENABLE IDLE SASL-IR listed AUTH=LOGINA0001 have AUTH=PLAIN LOGIN-REFERRALS more ID post-login LITERAL+ Pre-login NAMESPACE OK capabilities IMAP4rev1
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: 995/tcp  open  ssl/pop3   Dovecot pop3d
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com,
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: |_pop3-capabilities: PIPELINING USER SASL(PLAIN LOGIN) RESP-CODES CAPA UIDL TOP STLS AUTH-RESP-CODE
[*] Nmap: 143/tcp  open  imap       Dovecot imapd
[*] Nmap: |_imap-capabilities: ENABLE IDLE SASL-IR listed AUTH=LOGINA0001 have AUTH=PLAIN LOGIN-REFERRALS more post-login ID capabilities LITERAL+ Pre-login NAMESPACE OK STARTTLS IMAP4rev1
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com, D
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: 443/tcp  open  ssl/http   Apache httpd
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com, D
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: | http-server-header:
[*] Nmap: |   Apache
[*] Nmap: |_  nginx/1.25.5
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: |_http-title: Certfied Hacker
[*] Nmap: 465/tcp  open  ssl/smtp   Exim smtpd 4.96.2
[*] Nmap: |_smtp-commands: Couldn't establish connection on port 465
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: 587/tcp  open  smtp       Exim smtpd 4.96.2
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com, D
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: | smtp-commands: box5331.bluehost.com Hello www.certifiedhacker.com [49.43.24.20], SIZE 52428800, 8BITMIME, PIPELINING, PIPECONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP
[*] Nmap: |_ Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA BDAT NOOP QUIT RSET HELP
[*] Nmap: 993/tcp  open  ssl/imap   Dovecot imapd
[*] Nmap: |_imap-capabilities: ENABLE IDLE SASL-IR listed AUTH=LOGINA0001 have AUTH=PLAIN LOGIN-REFERRALS more ID post-login LITERAL+ Pre-login NAMESPACE OK capabilities IMAP4rev1
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com, D
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: |_Not valid after:  2024-12-30T15:01:15
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: 995/tcp  open  ssl/pop3   Dovecot pop3d
[*] Nmap: | ssl-cert: Subject: commonName=webmail.certifiedhacker.com
[*] Nmap: | Subject Alternative Name: DNS:autodiscover.certifiedhacker.com, DNS:certifiedhacker.com, DNS:cpanel.certifiedhacker.com, DNS:cpcalendars.certifiedhacker.com, DNS:cpcontacts.certifiedhacker.com, D
, DNS:mail.uyr.fvr.mybluehost.me, DNS:uyr.fvr.mybluehost.me, DNS:webdisk.certifiedhacker.com, DNS:webmail.certifiedhacker.com, DNS:www.certifiedhacker.com, DNS:www.uyr.fvr.mybluehost.me
[*] Nmap: | Not valid before: 2024-10-01T15:01:16
[*] Nmap: Network Distance: 2 hops
[*] Nmap: Service Info: OS: Linux; CPE: cpe:/o:redhat:enterprise_linux:7
[*] Nmap: TRACEROUTE (using port 80/tcp)
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   0.14 ms 192.168.153.2
[*] Nmap: 2   0.15 ms box5331.bluehost.com (162.241.216.11)
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 228.30 seconds
```

The db_nmap start the Nmap scan and the results would than be stored automatically in our database.

10. Type ***services*** or ***db_services*** to get the whole list of the services running on the host.

```
msf6 > services
Services
========

host            port  proto  name        state  info
----            ----  -----  ----        -----  ----
162.241.216.11  21    tcp    ftp         open   Pure-FTPd
162.241.216.11  22    tcp    ssh         open   OpenSSH 7.4 protocol 2.0
162.241.216.11  26    tcp    smtp        open   Exim smtpd 4.96.2
162.241.216.11  53    tcp    domain      open   ISC BIND 9.11.4-P2 RedHat Enterprise Linux 7
162.241.216.11  80    tcp    http        open   Apache httpd
162.241.216.11  110   tcp    pop3        open   Dovecot pop3d
162.241.216.11  143   tcp    imap        open   Dovecot imapd
162.241.216.11  443   tcp    ssl/http    open   Apache httpd
162.241.216.11  465   tcp    ssl/smtp    open   Exim smtpd 4.96.2
162.241.216.11  587   tcp    smtp        open   Exim smtpd 4.96.2
162.241.216.11  993   tcp    ssl/imap    open   Dovecot imapd
162.241.216.11  995   tcp    ssl/pop3    open   Dovecot pop3d
162.241.216.11  2222  tcp    ssh         open   OpenSSH 7.4 protocol 2.0
162.241.216.11  3306  tcp    mysql       open   MySQL 5.7.23-23
162.241.216.11  5432  tcp    postgresql  open   PostgreSQL DB
```

11. Load the **scanner/smb/smb_version** module: ***use scanner/smb/smb_version***
12. Type ***show options*** to see the configuration.

```
msf6 > use scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/ba
                                       sics/using-metasploit.html
   RPORT                     no        The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.
```

13. Set the ***RHOSTS*** to the target and ***THREADS*** to ***100 set RHOSTS 162.241.216.0-255***

set THREADS 100

```
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 162.241.216.0-255
RHOSTS => 162.241.216.0-255
msf6 auxiliary(scanner/smb/smb_version) > set THREADS 100
THREADS => 100
```

14. Type ***run*** to launch the module.

set

```
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 162.241.216.0-255:    - Scanned  27 of 256 hosts (10% complete)
[*] 162.241.216.0-255:    - Scanned  55 of 256 hosts (21% complete)
[*] 162.241.216.0-255:    - Scanned  77 of 256 hosts (30% complete)
[*] 162.241.216.0-255:    - Scanned 103 of 256 hosts (40% complete)
[*] 162.241.216.0-255:    - Scanned 130 of 256 hosts (50% complete)
[*] 162.241.216.0-255:    - Scanned 155 of 256 hosts (60% complete)
[*] 162.241.216.0-255:    - Scanned 184 of 256 hosts (71% complete)
[*] 162.241.216.0-255:    - Scanned 205 of 256 hosts (80% complete)
[*] 162.241.216.0-255:    - Scanned 232 of 256 hosts (90% complete)
[*] 162.241.216.0-255:    - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

This module will enumerate every open TCP services using a raw SYN scan.

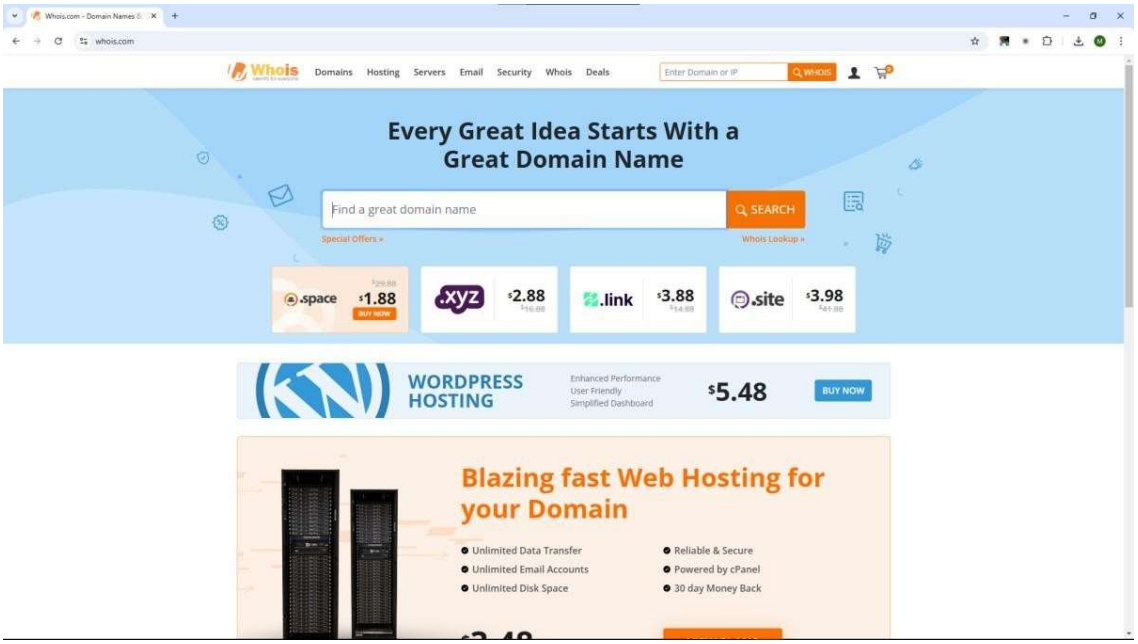
15. Now type ***hosts*** and observe the field **os_flavor** of the host you scanned in the subnet.

```
msf6 auxiliary(scanner/smb/smb_version) > hosts

Hosts
=====

address         mac  name                os_name   os_flavor  os_sp  purpose  info  comments
-------         ---  ----                -------   ---------  -----  -------  ----  --------
162.241.216.11       box5331.bluehost.com  embedded                  device
```

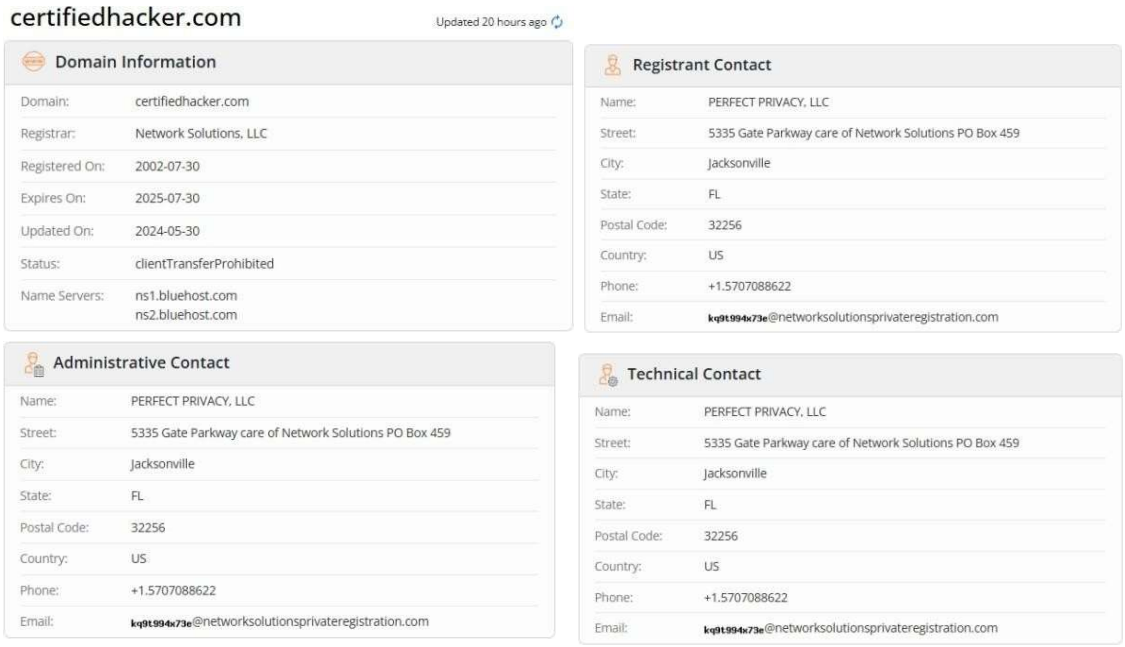
E. DNS WhoIsLookup (Web Based)

WHOIS helps to gain information regarding domain name, ownership information. IP Address, Netblock data, Domain Name Servers and other information's. Regional Internet Registries (RIR) maintain WHOIS database. WHOIS lookup helps to find out who is behind the target domain name.

1. Go to the URL https://www.whois.com/



2. A search of Target Domain.



Raw Whois Data

```
Domain Name: CERTIFIEDHACKER.COM
Registry Domain ID: 88849376_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.networksolutions.com
Registrar URL: http://networksolutions.com
Updated Date: 2024-08-22T07:51:37Z
Creation Date: 2002-07-30T00:32:00Z
Registrar Registration Expiration Date: 2025-07-30T00:32:00Z
Registrar: Network Solutions, LLC
Registrar IANA ID: 2
Reseller:
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: PERFECT PRIVACY, LLC
Registrant Organization:
Registrant Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Registrant City: Jacksonville
Registrant State/Province: FL
Registrant Postal Code: 32256
Registrant Country: US
Registrant Phone: +1.5707088622
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: kq0t994x73e@networksolutionsprivateregistration.com
Registry Admin ID:
Admin Name: PERFECT PRIVACY, LLC
Admin Organization:
Admin Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Admin City: Jacksonville
Admin State/Province: FL
Admin Postal Code: 32256
Admin Country: US
Admin Phone: +1.5707088622
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: kq0t994x73e@networksolutionsprivateregistration.com
Registry Tech ID:
Tech Name: PERFECT PRIVACY, LLC
Tech Organization:
Tech Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Tech City: Jacksonville
Tech State/Province: FL
Tech Postal Code: 32256
Tech Country: US
Tech Phone: +1.5707088622
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: kq0t994x73e@networksolutionsprivateregistration.com
Name Server: NS1.BLUEHOST.COM
Name Server: NS2.BLUEHOST.COM
DNSSEC: unsigned
Registrar Abuse Contact Email: domain.operations@web.com
Registrar Abuse Contact Phone: +1.8777228662
URL of the ICANN WHOIS Data Problem Reporting System: http://wdprs.internic.net/
>>> Last update of WHOIS database: 2024-10-29T16:39:12Z <<<
```

WHOIS Lookup Result shows complete domain profile, including:

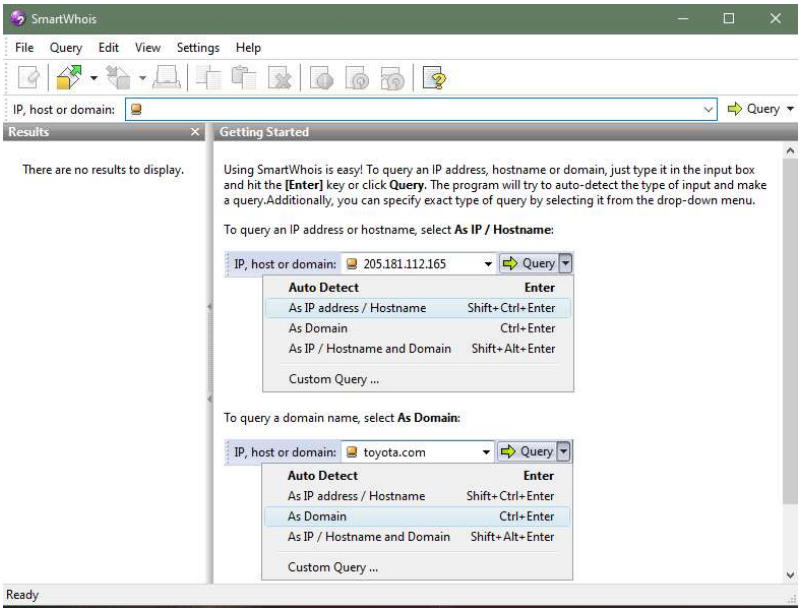
i. Registrant information
ii. Registrant Organization
iii. Registrant Country
iv. Domain name server information
v. IP Address
vi. IP location
vii. ASN
viii. Domain Status
ix. WHOIS history
x. IP history,
xi. Registrar history
xii. Hosting history

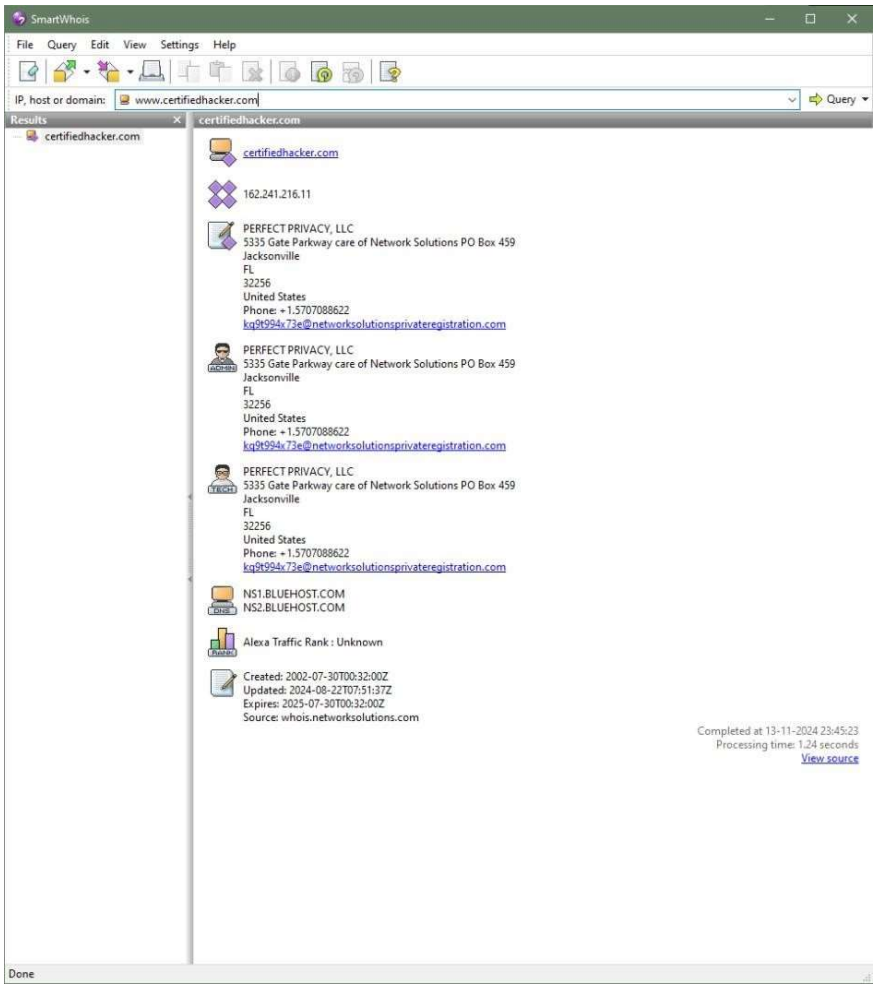
It also includes other information such as Email and postal address of registrar & admin along with contact details. You can go to https://whois.domaintools.com can enter the targeted URL

for WhoIsLookup information.

F. Smart WhoIs

You can download software “SmartWhois” from www.tamos.com

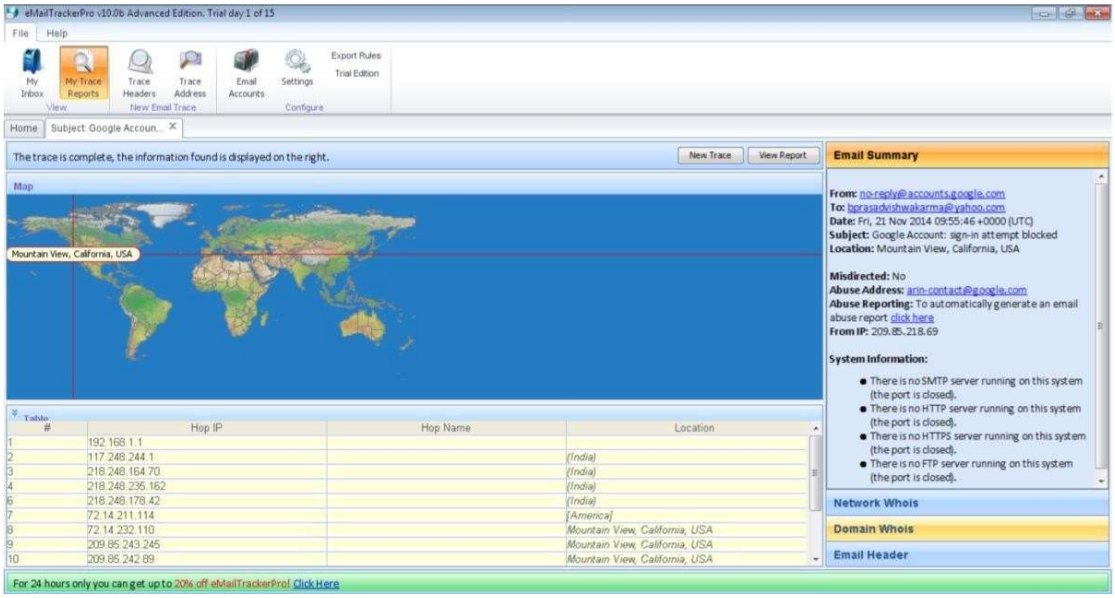




G. eMailTracker Pro

eMailTrackerPro is a Windows based email tracker that can be used to monitor employees, senders and recipients. This powerful tool can be used in conjunction with other programs such as Windows Nuke (also known as SpamWasher) to quickly identify where a computer has been and how it has been used.

Click on Trace Headers/Trace email address and enter the Message Header and click Okay. The Status of the Trace will be shown inside Trace Reports.



Practical No. 2

Aim: Perform the scanning of the networks using the tools:

A. **Hping2 for DoS attack (Kali Linux)**
B. **Advanced IP Scanner**
C. **Angry IP Scanner**
D. **Masscan (Kali Linux)**
E. **Scanning open ports of the system using CurrPorts**
F. **Create a TCP, UDP or SNMP packet using Colasoft Packet Builder**
G. **TheDude**

Scanning of the Network

Network Scanning refers to a set of procedures performed to identify hosts, ports, and services running in a network.

The purpose of network scanning is as follows:

i. Recognize available UDP and TCP network services running on the targeted hosts.
ii. Recognize filtering systems between the user and the targeted hosts.
iii. Determine the operating systems (OSs) in use by assessing IP responses.
iv. Evaluate the target host's TCP sequence number predictability to determine sequence prediction attack and TCP spoofing.

A. Hping2 / Hping3

Hping is a command-line TCP/IP packet assembler and analyzer tool that is used to send customized TCP/IP packets and display the target reply as ping command display the ICMP Echo Reply packet from targeted host. Hping can also handle fragmentation, arbitrary packets body, and size and file transfer. It supports TCP, UDP, ICMP and RAW-IP protocols.

Using Hping, the following parameters can be performed:

i. Test firewall rules.
ii. Advanced port scanning.
iii. Testing net performance.
iv. Path MTU discovery.
v. Transferring files between even fascist firewall rules.
vi. Traceroute-like under different protocols.
vii. Remote OS fingerprinting & others

1. Use ***hping3 -h*** to show all the commands.

2. Perform a basic hping3 scan using ***hping3 -c <packet_count> <Target IP address>***

   -c stands for packet count.

```
┌──(sms㉿kali)-[~]
└─$ sudo hping3 -c 3 www.certifiedhacker.com
[sudo] password for sms:
HPING www.certifiedhacker.com (eth0 162.241.216.11): NO FLAGS are set, 40 headers + 0 data bytes

--- www.certifiedhacker.com hping statistic ---
3 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

3. Create an ACK packet by typing the following command: ***hping3 -A <Target IP address>***

   -A represents ACK flag.

```
┌──(sms㉿kali)-[~]
└─$ sudo hping3 -A www.certifiedhacker.com
HPING www.certifiedhacker.com (eth0 162.241.216.11): A set, 40 headers + 0 data bytes
len=46 ip=162.241.216.11 ttl=128 id=33604 sport=0 flags=R seq=0 win=32767 rtt=5.4 ms
len=46 ip=162.241.216.11 ttl=128 id=33605 sport=0 flags=R seq=1 win=32767 rtt=3.4 ms
len=46 ip=162.241.216.11 ttl=128 id=33606 sport=0 flags=R seq=2 win=32767 rtt=2.5 ms
len=46 ip=162.241.216.11 ttl=128 id=33607 sport=0 flags=R seq=3 win=32767 rtt=2.0 ms
^C
--- www.certifiedhacker.com hping statistic ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2.0/3.3/5.4 ms
```

4. Create a SYN scan against different ports using ***hping3 –scan 1-3000 -S <Target IP address>***

   --scan parameter defines the port range to scan.

   -S represents SYN flag.

```
┌──(sms㉿kali)-[~]
└─$ sudo hping3 --scan 1-3000 -S www.certifiedhacker.com
Scanning www.certifiedhacker.com (162.241.216.11), port 1-3000
3000 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   22 ssh        : .S..A... 128 19075 64240    46
   21 ftp        : .S..A... 128 19331 64240    46
  587 submission : .S..A... 128 23427 64240    46
   26            : .S..A... 128 23683 64240    46
   53 domain     : .S..A... 128 23939 64240    46
  143 imap2      : .S..A... 128 24195 64240    46
  993 imaps      : .S..A... 128 38276 64240    46
  995 pop3s      : .S..A... 128 38532 64240    46
   80 http       : .S..A... 128 41092 64240    46
  110 pop3       : .S..A... 128 41348 64240    46
  465 submissions: .S..A... 128 41604 64240    46
  443 https      : .S..A... 128 41860 64240    46
All replies received. Done.
```

5. Create a packet with FIN, URG and PSH flag sets using ***hping3 -F -P -U <Target IP address>***

   -F represents FIN flag.

   -P represents PSH flag.

   -U represents URG flag.

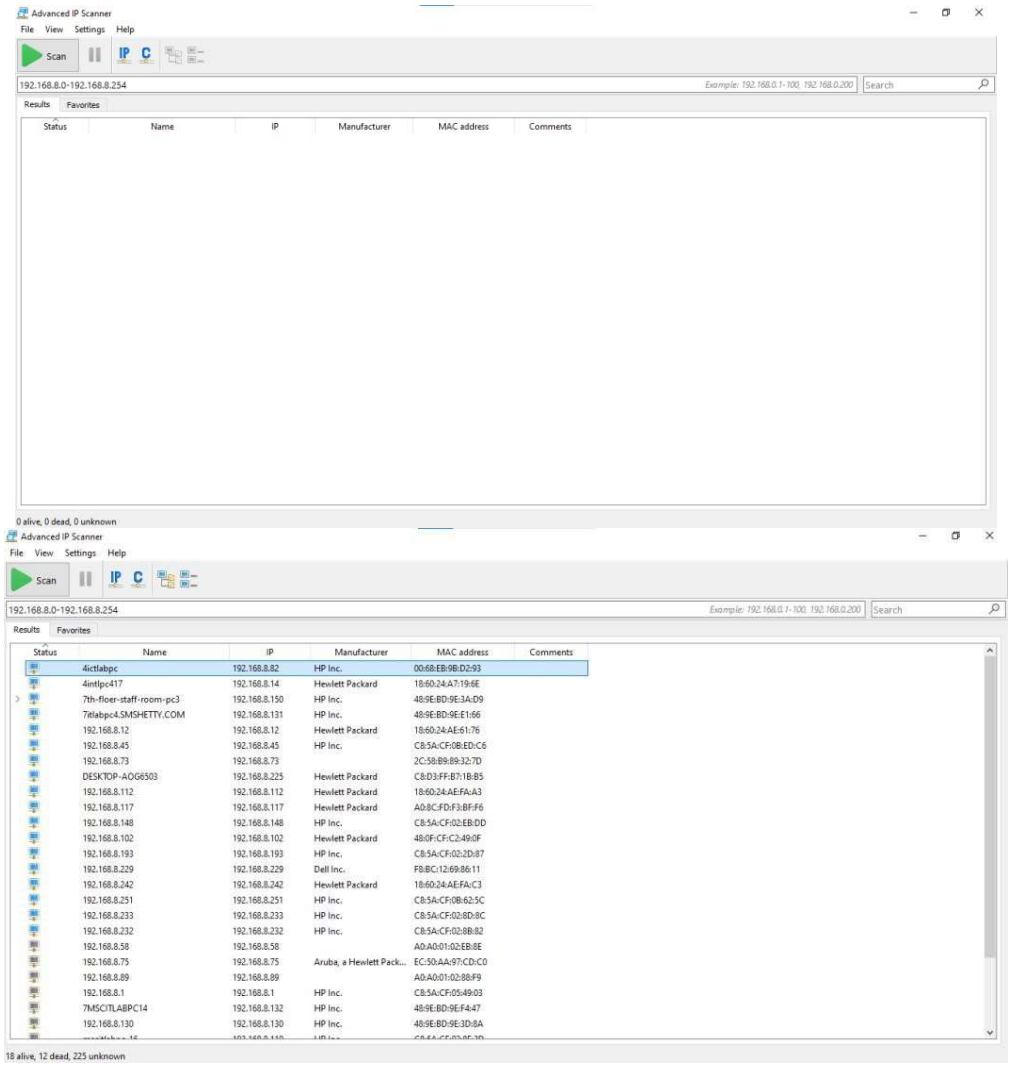
```
┌──(sms㉿kali)-[~]
└─$ sudo hping3 -F -P -U www.certifiedhacker.com
HPING www.certifiedhacker.com (eth0 162.241.216.11): FPU set, 40 headers + 0 data bytes
```

6. Create a packet to overwhelms the target's TCP stack by sending a large number of SYN requests without completing the handshake using ***hping3 --flood -S -d 2000 -a <source ip> <dest ip>***

```
┌──(sms㉿kali)-[~]
└─$ sudo hping3 --flood -S -d 2000 -a 192.168.153.128 www.certifiedhacker.com
HPING www.certifiedhacker.com (eth0 162.241.216.11): S set, 40 headers + 2000 data bytes
hping in flood mode, no replies will be shown
^C
--- www.certifiedhacker.com hping statistic ---
720263 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

B. Advanced IP Scanner

Advanced IP Scanner is a fast and powerful network scanner with a user-friendly interface. In seconds, Advanced IP Scanner can locate all computers on your wired or wireless local network and scan their ports. The program provides easy access to various network resources such as HTTP, HTTPS, FTP, and shared folders.

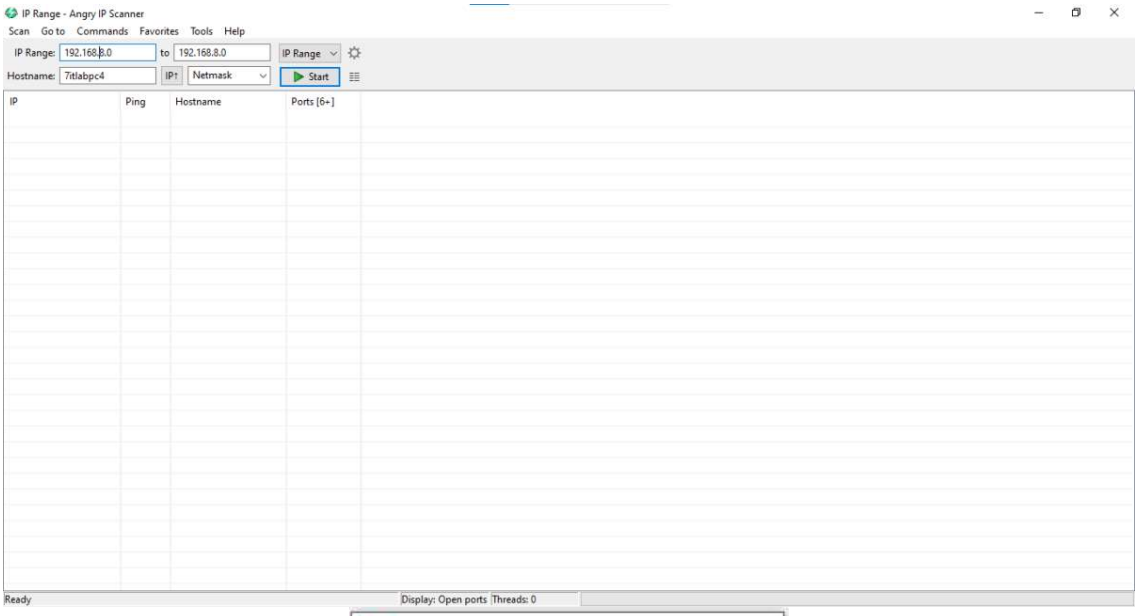


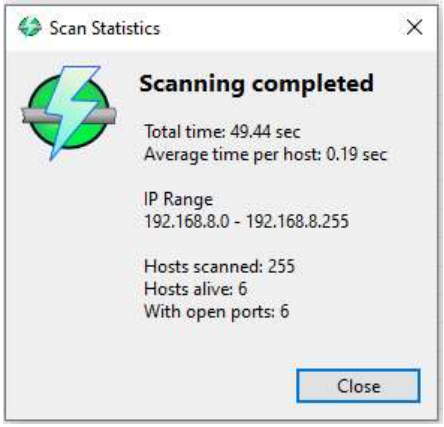
C. Angry IP Scanner

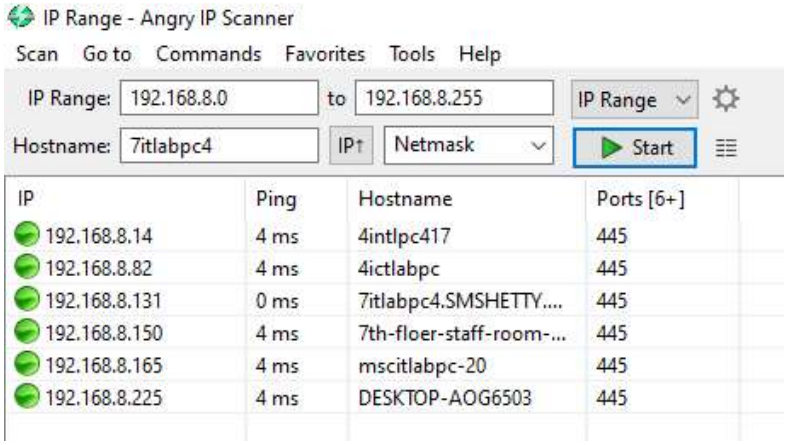
Angry IP Scanner (or simply ipscan) is an open-source and cross-platform network scanner designed to be fast and simple to use. It scans IP addresses and ports as well as has many other features.

It is widely used by network administrators and just curious users around the world, including large and small enterprises, banks, and government agencies.

It runs on Linux, Windows, and Mac OS X, possibly supporting other platforms as well.


IP Range - Angry IP Scanner
Scan Go to Commands Favorites Tools Help
IP Range: 192.168.8.0 to 192.168.8.0 IP Range
Hostname: 7itlabpc4 IP↑ Netmask Start
IP Ping Hostname Ports [6+]
Ready
Display: Open ports Threads: 0


Scan Statistics
Scanning completed
Total time: 49.44 sec
Average time per host: 0.19 sec
IP Range
192.168.8.0 - 192.168.8.255
Hosts scanned: 255
Hosts alive: 6
With open ports: 6
Close


IP Range - Angry IP Scanner
Scan Go to Commands Favorites Tools Help
IP Range: 192.168.8.0 to 192.168.8.255 IP Range
Hostname: 7itlabpc4 IP↑ Netmask Start
IP Ping Hostname Ports [6+]
192.168.8.14 4 ms 4intlpc417 445
192.168.8.82 4 ms 4ictlabpc 445
192.168.8.131 0 ms 7itlabpc4.SMSHETTY.... 445
192.168.8.150 4 ms 7th-floer-staff-room-... 445
192.168.8.165 4 ms mscitlabpc-20 445
192.168.8.225 4 ms DESKTOP-AOG6503 445

D. Masscan (Kali Linux)

Masscan is TCP port scanner which transmits SYN packets asynchronously and produces results similar to nmap, the most famous port scanner. Internally, it operates more like scanrand, unicornscan, and ZMap, using asynchronous transmission. It's a flexible utility that allows arbitrary address and port ranges.

Scan for a selection of ports across a given subnet using command:

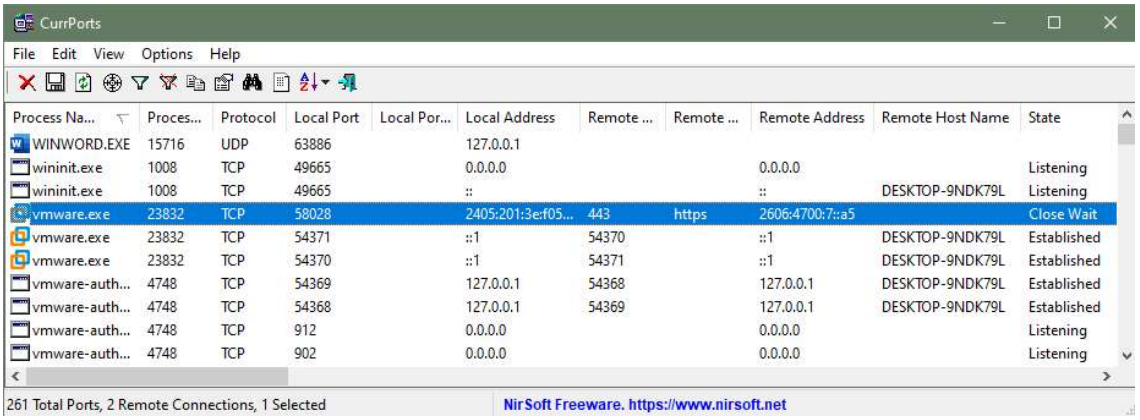
masscan -p <port> <subnet / IP address>

```
┌──(sms㉿kali)-[~]
└─$ sudo masscan -p 22,445,443,80 162.241.216.11
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2024-11-15 13:55:38 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [4 ports/host]
Discovered open port 80/tcp on 162.241.216.11
Discovered open port 22/tcp on 162.241.216.11
Discovered open port 443/tcp on 162.241.216.11
```

```
┌──(sms㉿kali)-[~]
└─$ sudo masscan -p 22,445,443,80 162.241.216.0/8
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2024-11-15 13:53:22 GMT
Initiating SYN Stealth Scan
Scanning 16777216 hosts [4 ports/host]
Discovered open port 22/tcp on 162.245.220.77
Discovered open port 443/tcp on 162.240.149.111
Discovered open port 80/tcp on 162.214.103.96
Discovered open port 443/tcp on 162.249.110.130
Discovered open port 80/tcp on 162.159.247.117
Discovered open port 22/tcp on 162.215.98.101
Discovered open port 80/tcp on 162.55.31.241
Discovered open port 80/tcp on 162.19.121.59
Discovered open port 22/tcp on 162.248.55.85
Discovered open port 80/tcp on 162.191.79.165
Discovered open port 80/tcp on 162.217.226.126
Discovered open port 80/tcp on 162.43.92.170
Discovered open port 80/tcp on 162.209.189.167
Discovered open port 22/tcp on 162.55.98.188
Discovered open port 443/tcp on 162.251.25.29
Discovered open port 22/tcp on 162.240.173.13
Discovered open port 80/tcp on 162.255.116.119
Discovered open port 80/tcp on 162.214.255.89
Discovered open port 443/tcp on 162.159.141.248
```

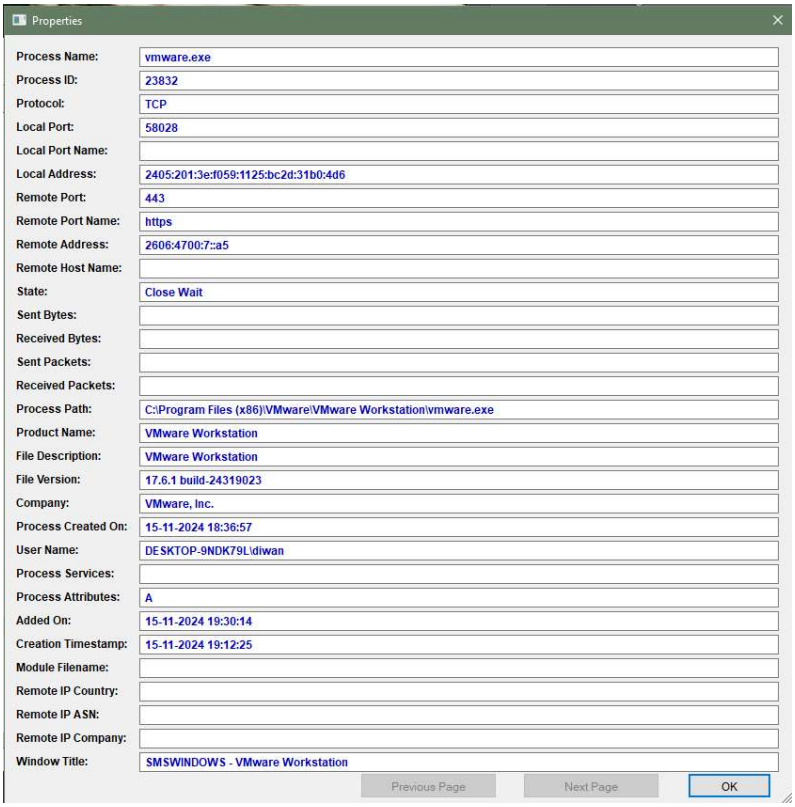
E. CurrPorts

CurrPorts is a lightweight and free utility for Windows that provides detailed information about open TCP/IP and UDP ports on a system. It displays active connections, including local and remote addresses, the process responsible for the connection, and the state of each port (e.g., listening, established, or closed). The tool is particularly useful for identifying unwanted or suspicious connections, as it highlights remote addresses and allows users to terminate connections directly from the interface. It also includes features for exporting data to a file, filtering displayed connections, and color-coding entries for easier analysis.



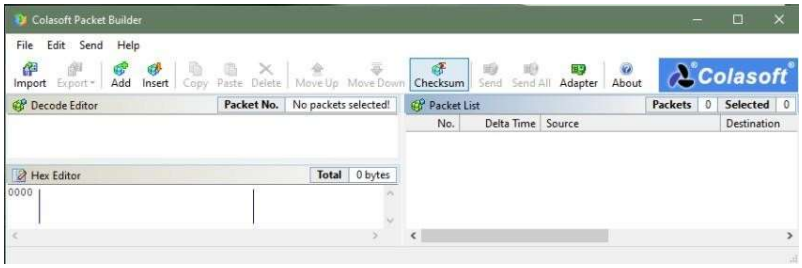
Process Na...	Proces...	Protocol	Local Port	Local Por...	Local Address	Remote ...	Remote ...	Remote Address	Remote Host Name	State
WINWORD.EXE	15716	UDP	63886		127.0.0.1					
wininit.exe	1008	TCP	49665		0.0.0.0			0.0.0.0		Listening
wininit.exe	1008	TCP	49665		::			::	DESKTOP-9NDK79L	Listening
vmware.exe	23832	TCP	58028		2405:201:3e:f05...	443	https	2606:4700:7::a5		Close Wait
vmware.exe	23832	TCP	54371		::1	54370		::1	DESKTOP-9NDK79L	Established
vmware.exe	23832	TCP	54370		::1	54371		::1	DESKTOP-9NDK79L	Established
vmware-auth...	4748	TCP	54369		127.0.0.1	54368		127.0.0.1	DESKTOP-9NDK79L	Established
vmware-auth...	4748	TCP	54368		127.0.0.1	54369		127.0.0.1	DESKTOP-9NDK79L	Established
vmware-auth...	4748	TCP	912		0.0.0.0			0.0.0.0		Listening
vmware-auth...	4748	TCP	902		0.0.0.0			0.0.0.0		Listening

For more detail, Click on a Process.



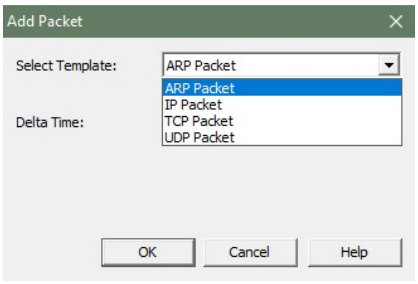
F. Colasoft Packet Builder

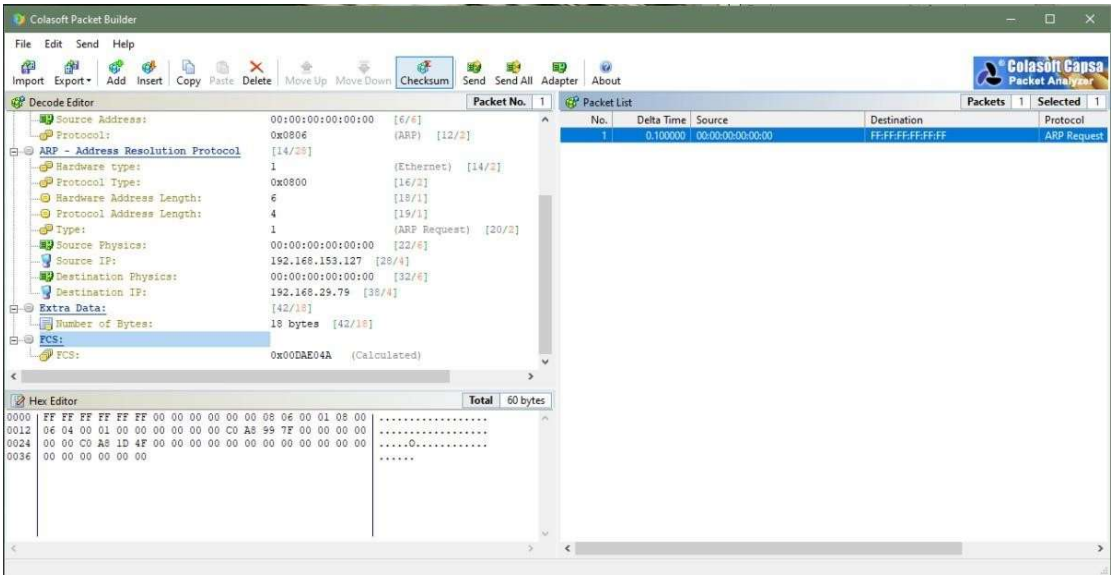
Colasoft Packet Builder enables creating custom network packets; users can use this tool to check their network protection against attacks and intruders. Colasoft Packet Builder includes a very powerful editing feature. Besides common HEX editing raw data, it features a Decoding Editor allowing users to edit specific protocol field values much easier. Customized Network packets can penetrate the network for attacks. Customization can also use to create fragmented packets. You can download the software from www.colasoft.com.



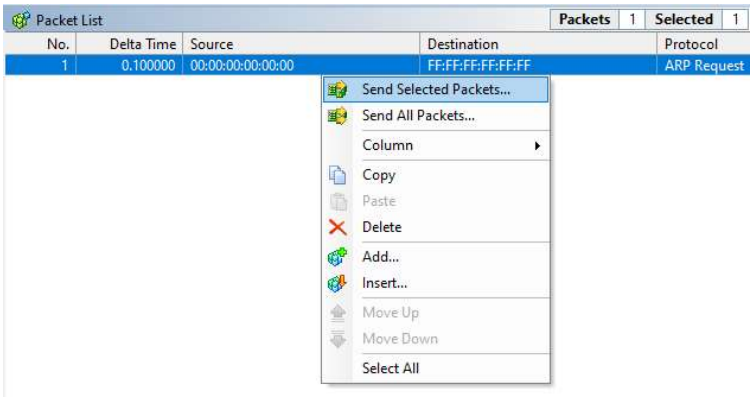
Colasoft packet builder offers Import and Export options for a set of packets. You can also add a new packet by clicking Add/button. Select the Packet type from the drop-down option. Available options are:

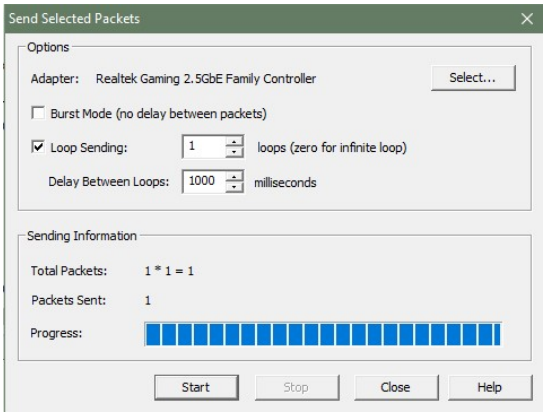
i. ARP Packet
ii. IP Packet
iii. TCP Packet
iv. UDP Packet

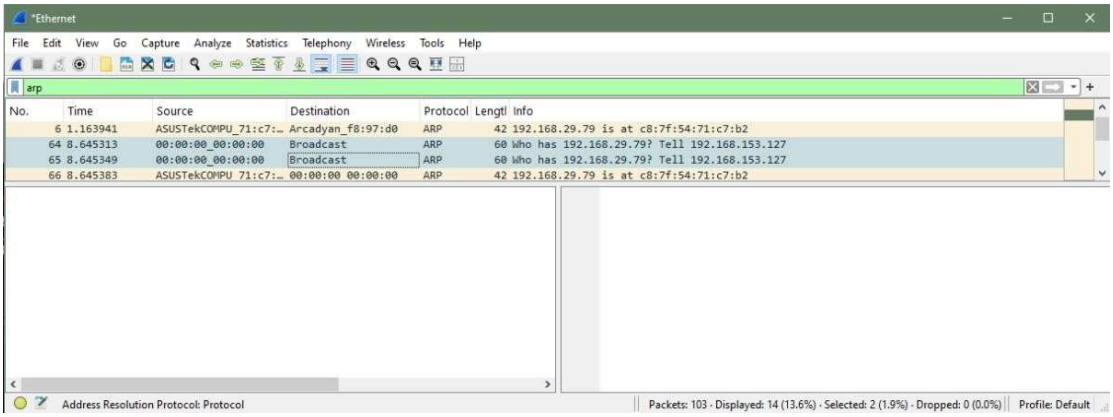




After Selecting the Packet Type, now you can customize the packet, Select the Network Adapter and Send it towards the destination.





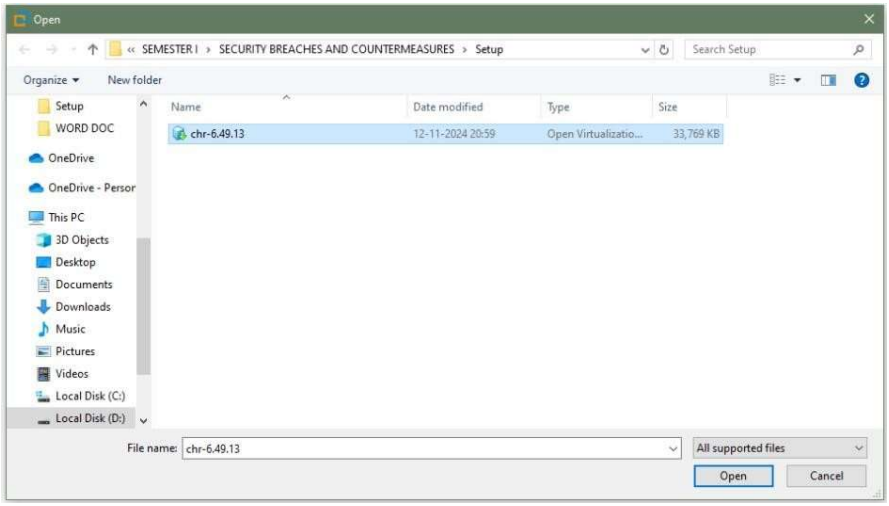


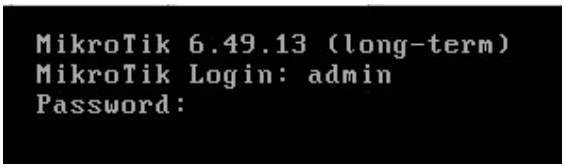
G. TheDude

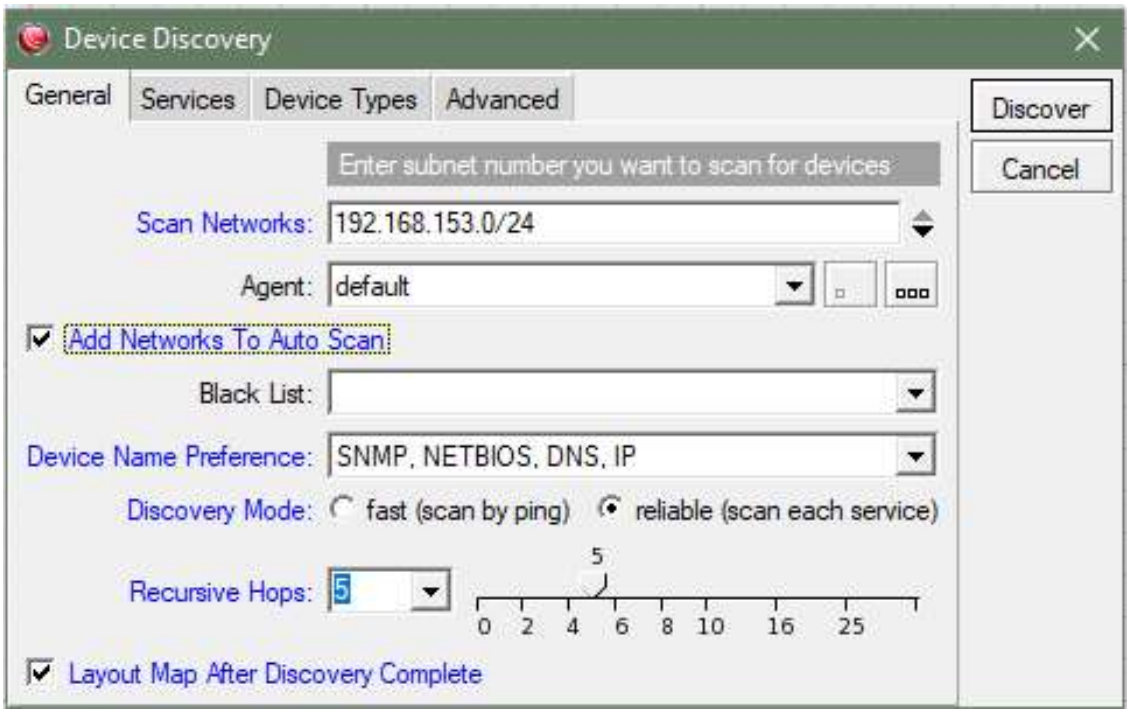
The Dude network monitor is a new application by MikroTik which can dramatically improve the way you manage your network environment. It will automatically scan all devices within specified subnets, draw and layout a map of your networks, monitor services of your devices and alert you in case some service has problems.

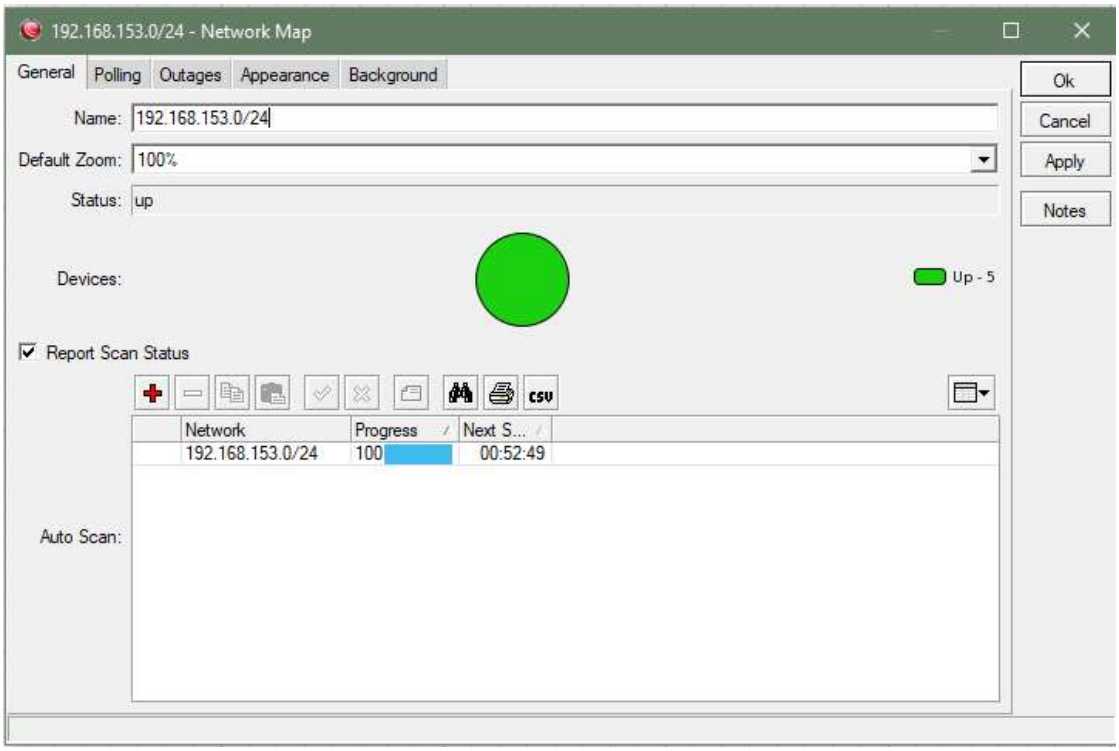
Main Features:

i. Auto network discovery and layout
ii. Discovers any type or brand of device
iii. Device, Link monitoring, and notifications
iv. Includes SVG icons for devices, and supports custom icons and backgrounds
v. Easy installation and usage
vi. Allows you to draw your own maps and add custom devices
vii. Supports SNMP, ICMP, DNS and TCP monitoring for devices that support it.
viii. Individual Link usage monitoring and graphs
ix. Direct access to remote control tools for device management
x. Supports remote Dude server and local client






Device Discovery
General
Services
Device Types
Advanced
Discover
Cancel
Enter subnet number you want to scan for devices
Scan Networks:
192.168.153.0/24
Agent:
default
Add Networks To Auto Scan
Black List:
Device Name Preference:
SNMP, NETBIOS, DNS, IP
Discovery Mode:
fast (scan by ping)
reliable (scan each service)
Recursive Hops:
5
0 2 4 6 8 10 16 25
Layout Map After Discovery Complete

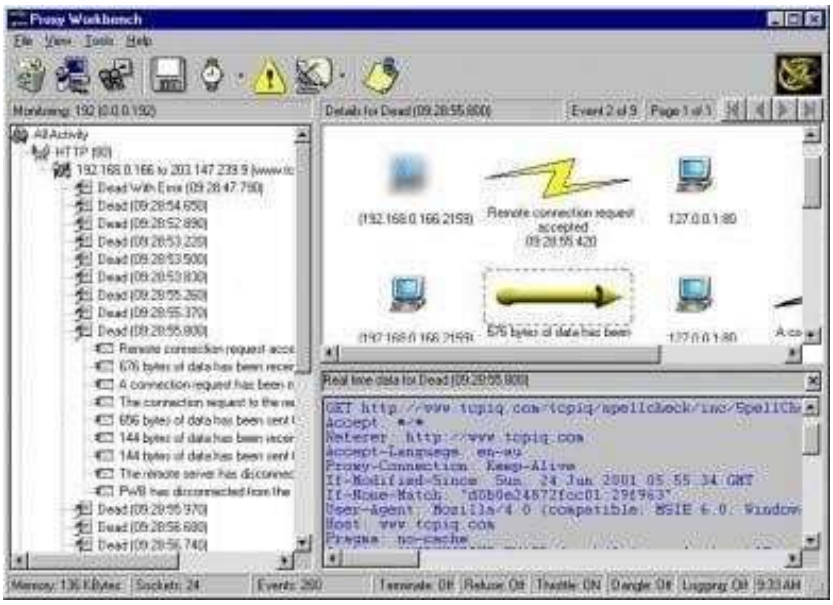

192.168.153.0/24 - Network Map
General
Polling
Outages
Appearance
Background
Ok
Cancel
Apply
Notes
Name:
192.168.153.0/24
Default Zoom:
100%
Status:
up
Devices:
Up - 5
Report Scan Status
csv
Network
Progress
Next S...
192.168.153.0/24
100
00:52:49
Auto Scan:

Practical No. 3

A. Use Proxy Workbench Tool

Proxy Workbench is a unique proxy server - ideal for developers, security experts, and trainers that displays data in real time. You can actually see the data flowing between your e-mail client and the e-mail server, web browser and web server or even analyze FTP in both Passive and Active modes. In addition, the 'pass through' protocol handler enables analysis of protocols where the server does not readily change.

The best feature is the animated connection diagram that graphically represents the history of each socket connection and allows you to drill into the finest of detail. This animation can even be exported to HTML and saved to the web!

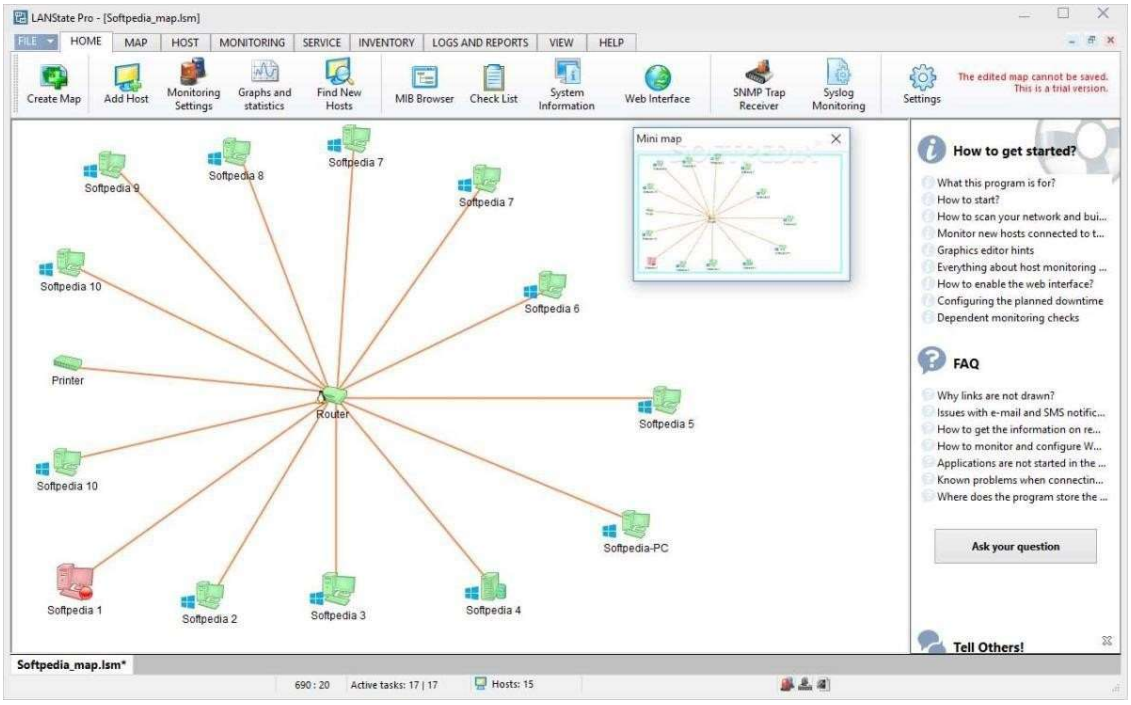


B. Network Discovery

Network Discovery is used to identify, map and monitor devices on a network. These tools are crucial for managing and ensuring the reliability of networks in IT environments.

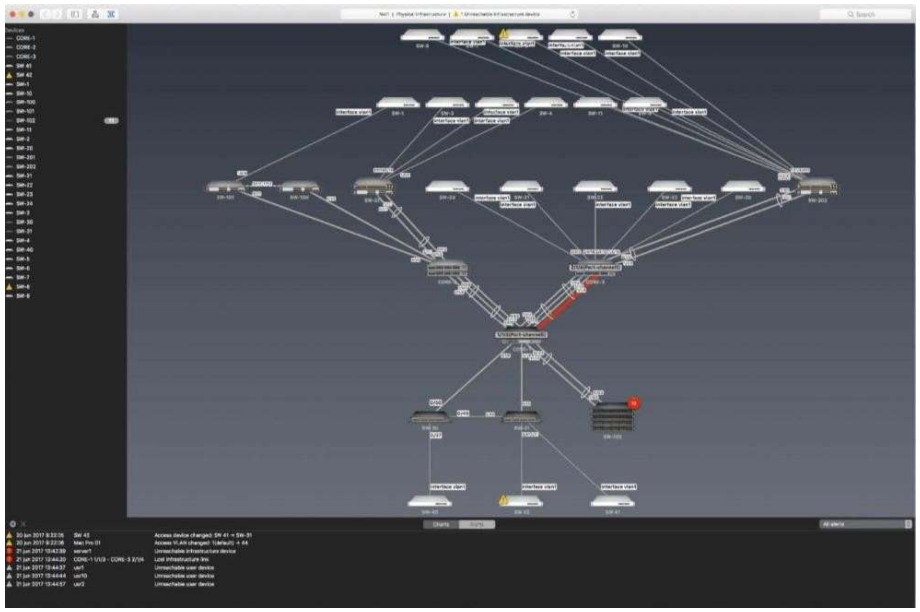
a. LANState Pro

LANState is a simple network topology mapping, host monitoring, and management program. Monitor the service availability. Manage servers, computers, switches, and other devices easier using the graphic map. Access devices' properties, RDP, web UI faster.



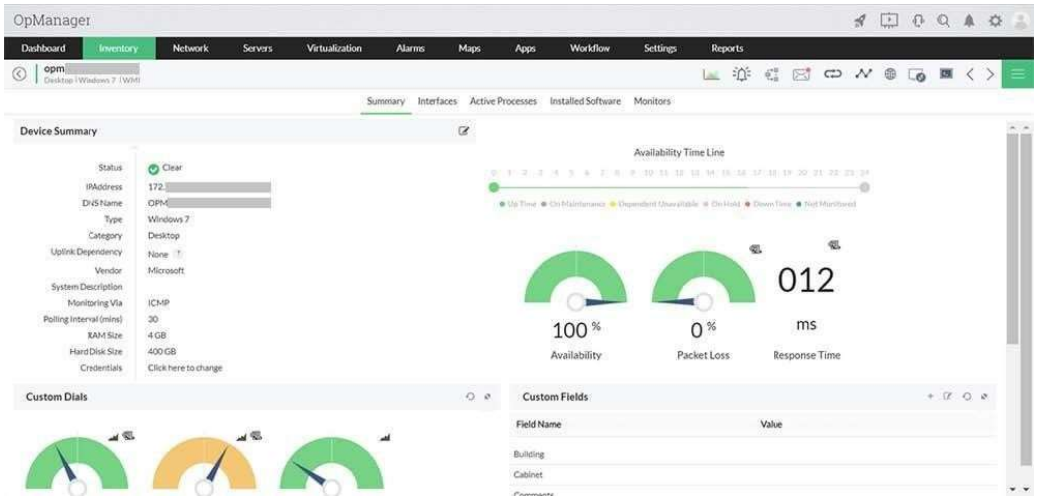
b. Network View

NetworkView is a network visualization tool that aims to provide a simple interface for the complex function involved in the discovery and monitoring of multi-vendor IP networks. With NetworkView you can get a quick overview of your network, whether it is a small office or a corporate network. Version 3 adds functionalities oriented to network management tasks.
NetworkView uses multiple methods such as ICMP, MDNS, SSDP, DNS, NetBIOS, SNMP MIB-2, Bridge MIB, LLDP, CPD and proprietary MIB's to discover devices and generates a graphical representation of your network. NetworkView generates views of both logical and physical network structure. Virtual structure representation is also displayed for wireless systems (Cisco, Aruba/Alcatel-Lucent and Fortinet).



c. OpManager

OpManager is an advanced network monitoring tool which offers fault management, supporting over WAN links, Router, Switch, VoIP & servers. It can also perform performance management.



Practical No. 4

Aim:

A. **Perform Enumeration using the following tools:**
   a. **Nmap**
   b. **NetBIOS**
   c. **Hyena**
   d. **SuperScan Software**
   e. **Wireshark**

B. **Perform Vulnerability Analysis using Nessus.**

A. Enumeration

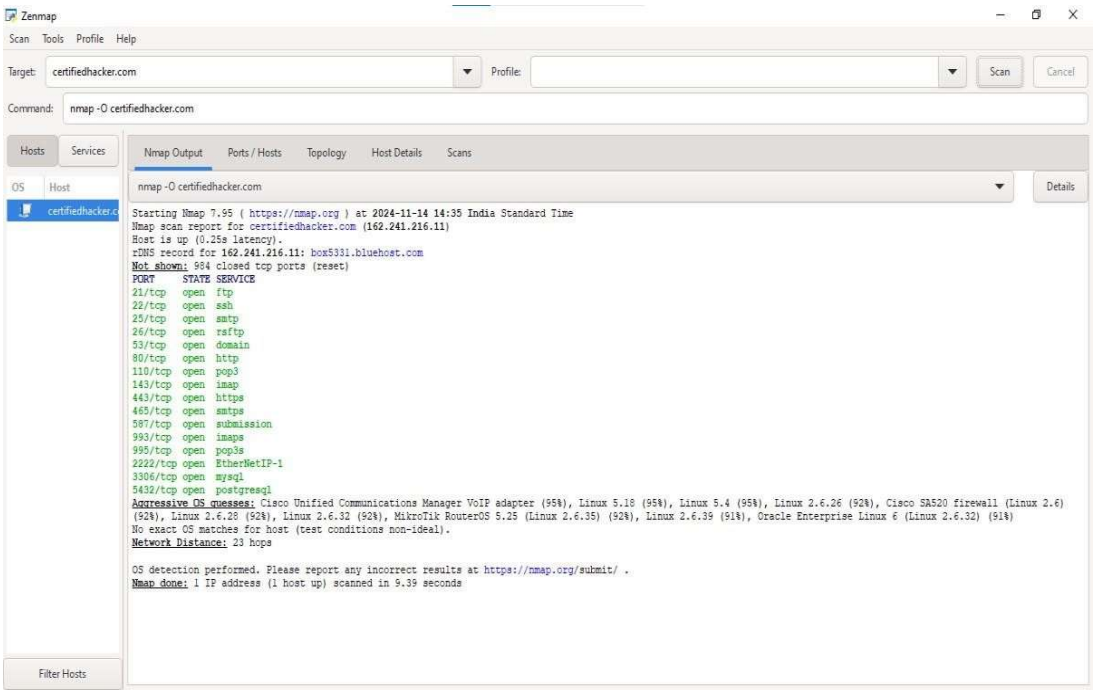
Enumeration is the process of extracting user names, machine names, network resources, shares, and services from a system, and its conducted in an intranet environment.

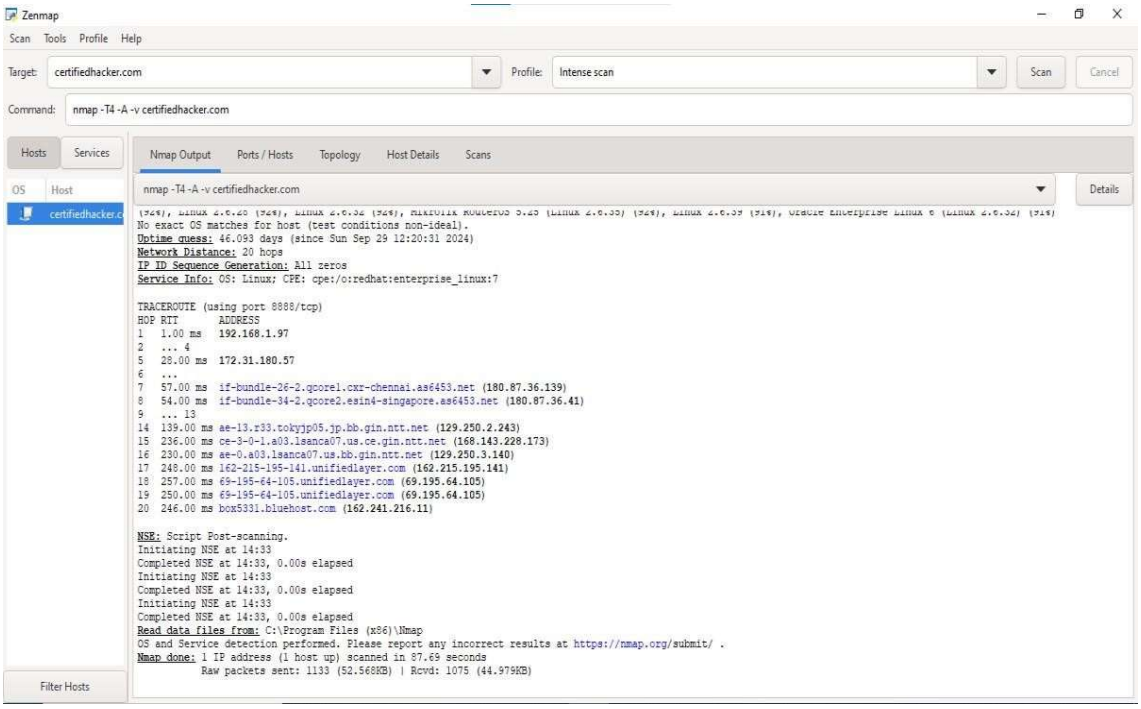
In this phase, the attacker creates an active connection to the system and performs directed queries to gain more information about the target. The gathered information is used to identify the vulnerabilities or weak points in system security and tries to exploit in the System gaining phase.

a. Nmap

NMAP, as we know, is a powerful networking tool which supports many features and commands. Operating System detection capability allows to send TCP and UDP packet and observe the response from the targeted host. A detailed assessment of this response bring some clues regarding nature of an operating system disclosing the type an OS.

To perform OS detection with nmap perform the following: ***nmap -O <ip address>***





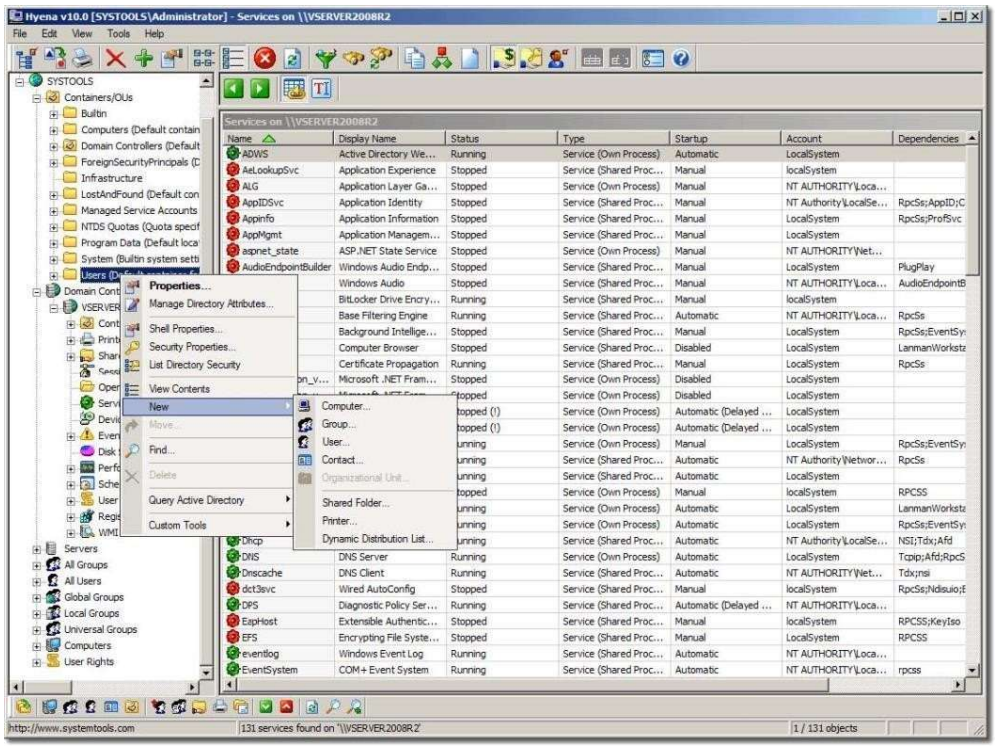
b. NetBIOS

NetBIOS stands for Network Basic Input Output System. It Allows computer communication over a LAN and allows them to share files and printers. NetBIOS names are used to identify network devices over TCP/IP (Windows).

```
┌──(ritik㉿ritik)-[~]
└─$ netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 ritik:45204            del12s05-in-f4.1e:https ESTABLISHED
tcp        0      0 ritik:49222            server-13-224-20-:https ESTABLISHED
tcp        0      0 ritik:34744            ec2-35-167-149-24:https ESTABLISHED
tcp        0      0 ritik:58126            ec2-35-161-6-128.:https ESTABLISHED
tcp        0      0 ritik:55236            104.18.32.68:http       TIME_WAIT
tcp        0      0 ritik:60936            98.203.120.34.bc.:https ESTABLISHED
tcp        0      0 ritik:43858            104.22.24.131:https     ESTABLISHED
tcp        0      0 ritik:37840            20.120.65.166:https     ESTABLISHED
tcp        0      0 ritik:46330            104.16.122.175:https    ESTABLISHED
udp        0      0 ritik:bootpc           WS-GFGDC01.ad.ge:bootps ESTABLISHED
raw6       0      0 [::]:ipv6-icmp         [::]:*                  7
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     197448   /run/user/1000/speech-dispatcher/speechd.sock
unix  2      [ ACC ]     STREAM     LISTENING     17408    /tmp/.X11-unix/X1
unix  2      [ ACC ]     STREAM     LISTENING     19999    @/tmp/.ICE-unix/1182
unix  3      [ ]         DGRAM      CONNECTED     14870    /run/systemd/notify
```

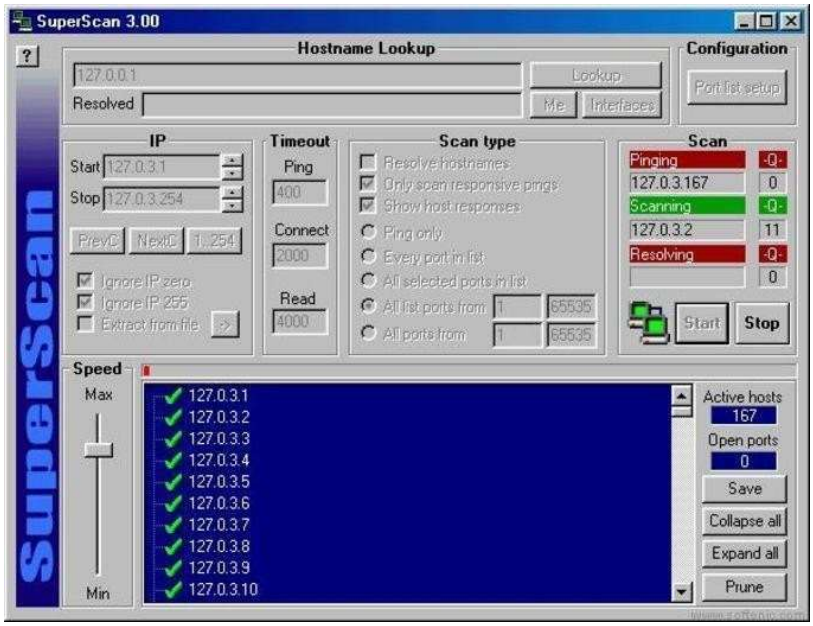
c. Hyena

Hyena is GUI based, NetBIOS Enumeration tool that shows Shares, User login information and other related information



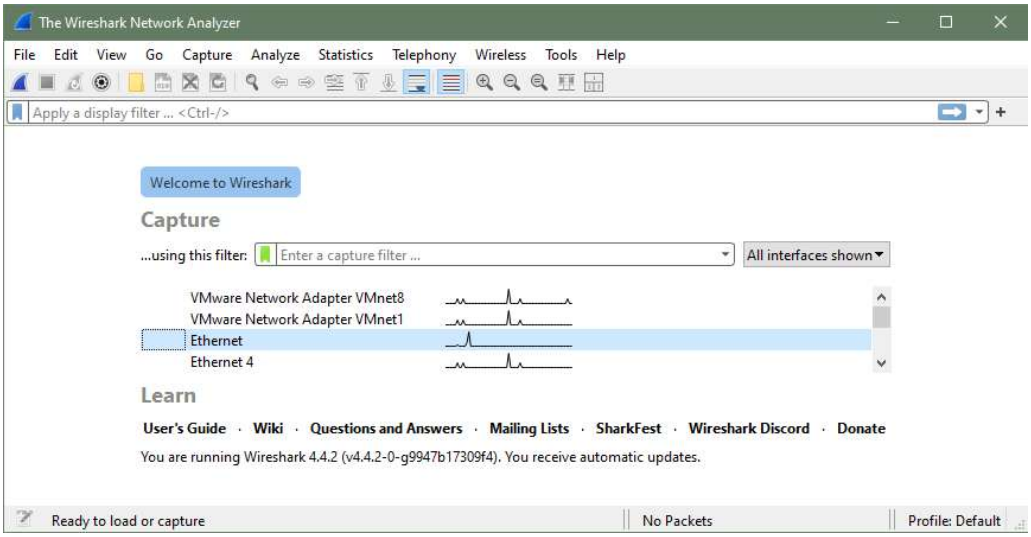
d. SuperScan Software

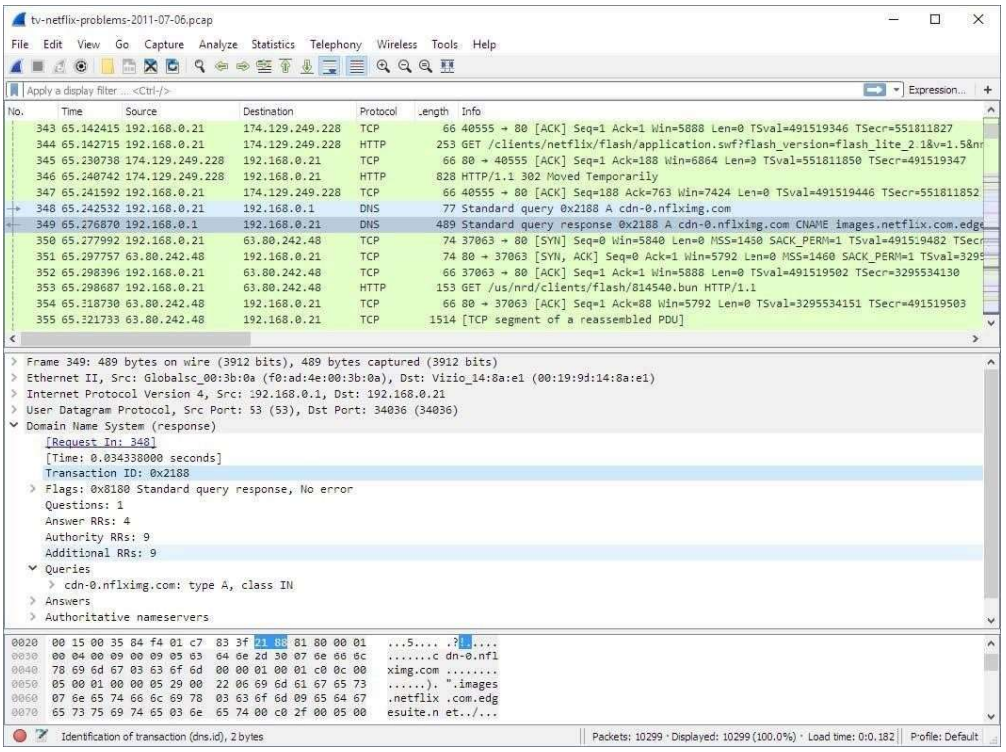
SuperScan is a multi-functional tool that will help you manage your network and make sure your connections and TCP ports are working as well as they should be. One of the best features or advantages of this tool is just how quickly it works. The scans are made very rapidly and faster than with most other scanning tools out there.



e. Wireshark

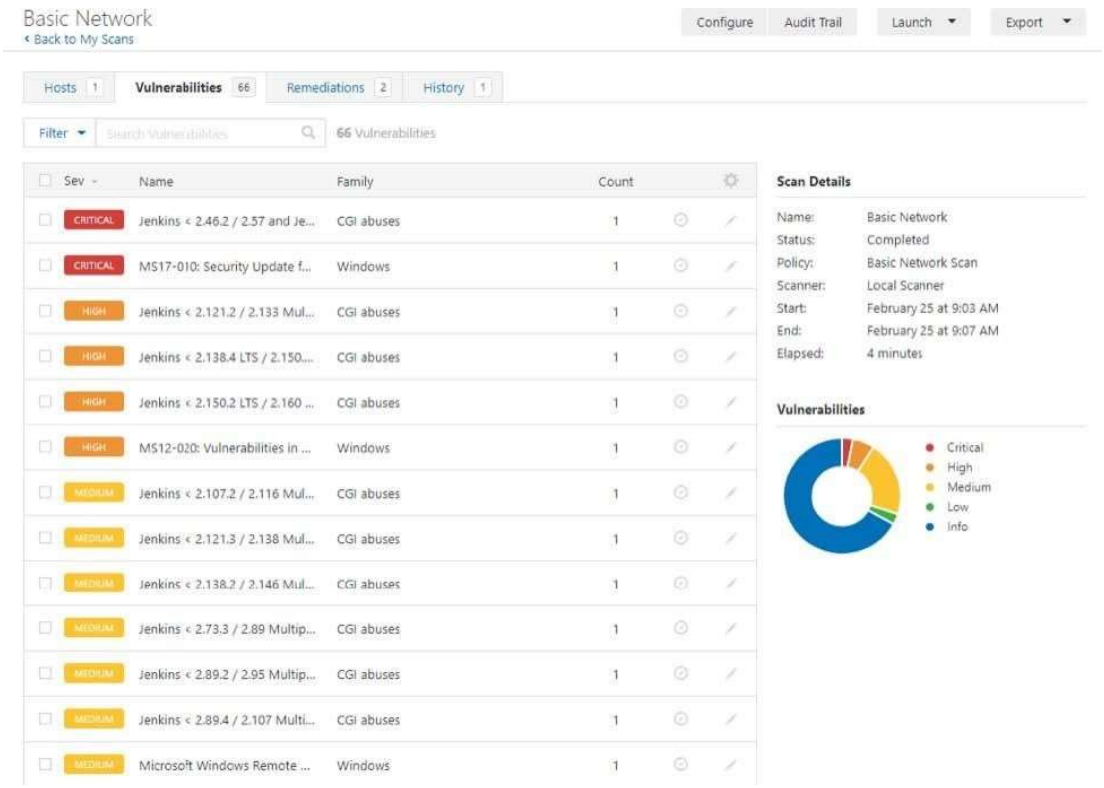
Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues





B. Vulnerability Analysis using Nessus.

Nessus is a proprietary vulnerability scanner developed by Tenable, Inc. Tenable.io is a subscription-based service. Tenable also contains what was previously known as Nessus Cloud, which used to be Tenable's Software-as-a-Service solution. Nessus is an open-source network vulnerability scanner that uses the Common Vulnerabilities and Exposures architecture for easy cross-linking between compliant security tools. In fact, Nessus is one of the many vulnerability scanners used during vulnerability assessments and penetration testing engagements, including malicious attacks. Nessus is a tool that checks computers to find vulnerabilities that hackers COULD exploit.



Practical No. 5

Aim: Perform the system hacking using the tools:

A. **Winrtgen**
B. **PWDump**
C. **Ophcrack**
D. **NTFS Stream Manipulation**
E. **ADS Spy**
F. **Quickstego**

System Hacking

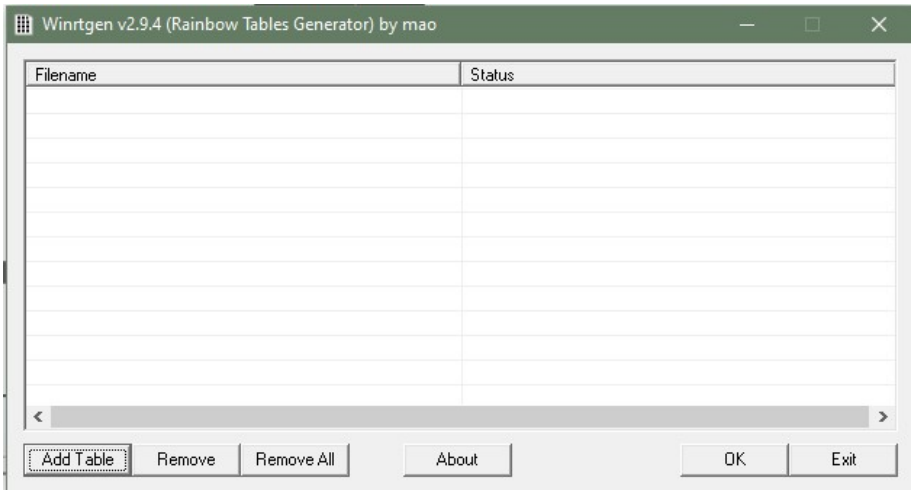
System hacking is the science of testing computers and network for vulnerabilities and harmful plug-ins. System hacking is itself a vast subject which consists of hacking the different software based technological systems such as laptops, desktops, etc. System hacking is defined as the compromise of computer systems and software to gain access to the target computer and steal or misuse their sensitive information. Here the malicious hacker exploits the weaknesses in a computer system or network to gain unauthorized access of its data or take illegal advantage of it.

A. WinRTGen

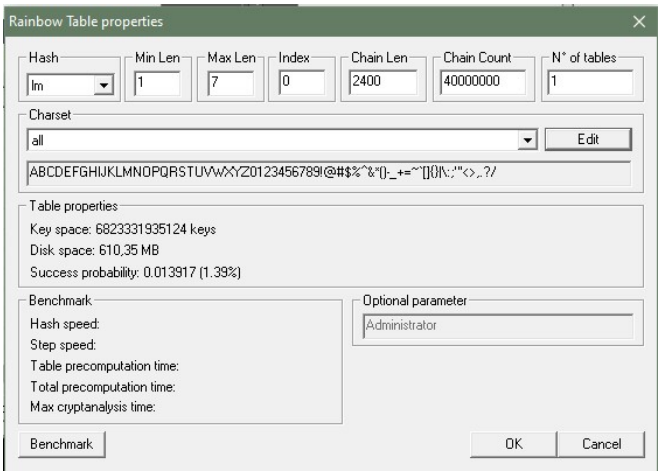
WinRTGen is a tool used to generate rainbow tables for password cracking. Rainbow tables are precomputed hash tables containing potential passwords mapped to their hash values. By consulting these tables, attackers can bypass the need for live brute-forcing by quickly looking up the hash of a password and matching it to a known plaintext, significantly speeding up the process of cracking hashed passwords.

It supports various hash types, such as LM, NTLM, MD5, SHA1, and many others, commonly used in Windows environments and some network protocols. Users can customize rainbow tables by setting parameters like password length, character set, chain length, and table size.

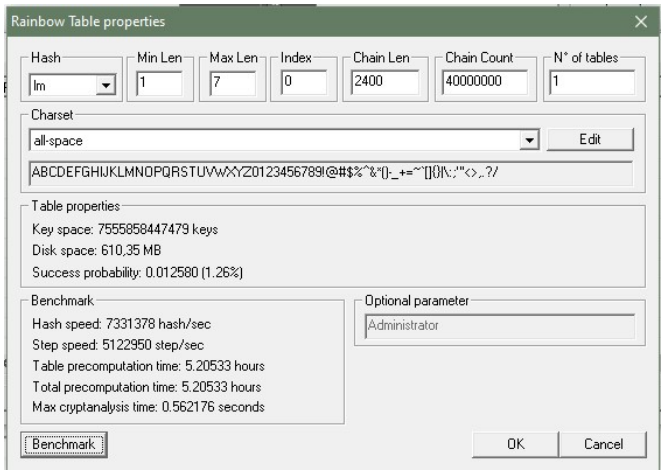
1. To generate rainbow tables first we will have to modify the properties of WinRTGen according to our need, and to do so Click on **Add Table**. After this, a new box will appear named **Rainbow Table Properties**.



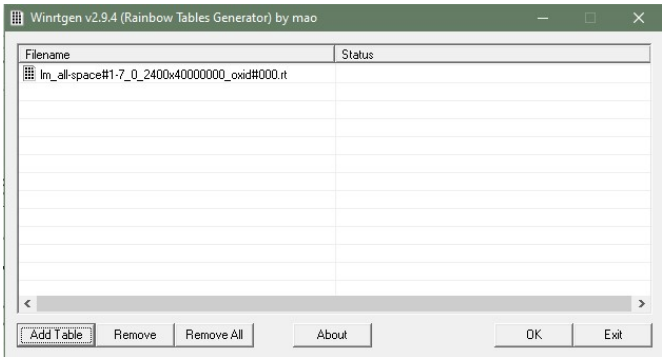
2. In the **Rainbow Table Properties** window we have the option to modify settings in order to generate rainbow tables according to our needs.



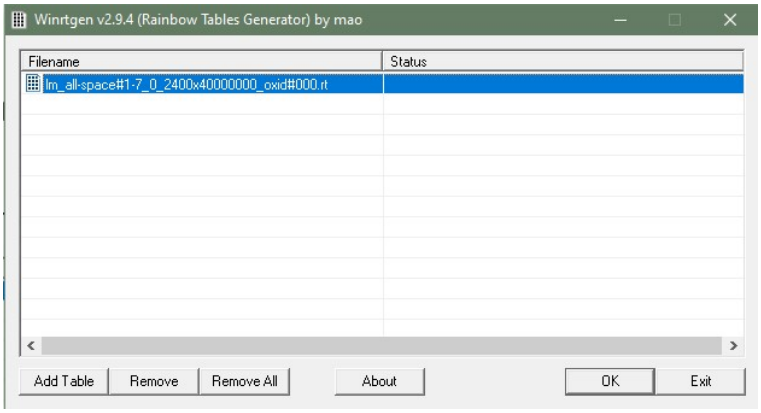
3. After assigning the values to the properties according to our needs click on **Benchmarks**. This will show the estimated time, Hash speed, Step speed, Table Pre- computing time, etc. that will be required to generate the Rainbow Table according to assigned properties.



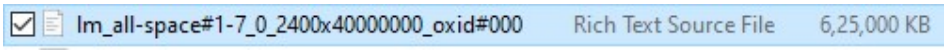
4. Click on **OK**. This will add the Rainbow Table to the queue in the main window of WinRTGen.



5. After this click on **Rainbow Table** you want to start processing and click **OK**, the WinRTGen will start generating a rainbow table.



6. After completion, this table will be saved to your WinRTGen Directory.



B. PWDump

PWDump is a collection of tools designed to extract hashed passwords from the Windows Security Account Manager (SAM) database and the NTDS file in Active Directory. The various PWDump versions generally function by accessing Windows' SAM files or the Active Directory database to retrieve password hashes, including LM and NTLM hashes. Some versions of PWDump (like pwdump7) employ unique techniques such as kernel-level access to bypass security restrictions that prevent unauthorized access to the SAM file. Running PWDump requires administrative privileges, and it typically interacts directly with low-level system files or database APIs, preserving system stability by not injecting code or creating new services.

1. Open a Command Prompt with administrator privileges. Navigate to the directory where PWDump is located.

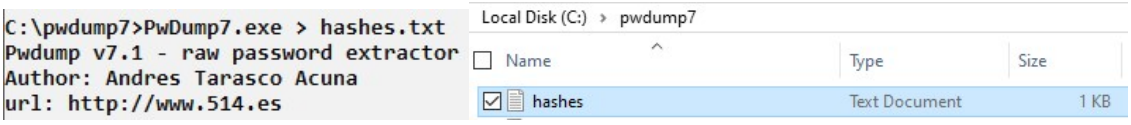
2. Use the command syntax for your specific PWDump version. For instance, in pwdump7: ***pwdump7.exe***

3. The output should display hashes for each user account on the system, in the format:

<Username>:<UserID>:<LM_Hash>:<NTLM_Hash>:::

```
C:\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:74C9C77ACB5DB5A649157356187707A9:836D97D699522704661D0EF503FCCB02:::
Guest:501:5BF1B83F437FBC4F40CF5A4A9D025FBF:8D213EF7B8812F0C98A876BAD07A2927:::
¶:503:0D88D0B103F312AA40C683FC2827D06F:0B5F87EE090E0A61F20188C2C90875AE:::
¶:504:0FB449B2442AF37628D43EA77D1B289A:ABDBE9294E8B5C56619BBF75E15C5FB7:::
diwan:1001:4534C36E23F91490B02C4AE105B79FC5:A67578C7C71E705E68C7DF1726859878:::
```

4. To save the output, redirect the output to a file for later use: pwdump7.exe > hashes.txt

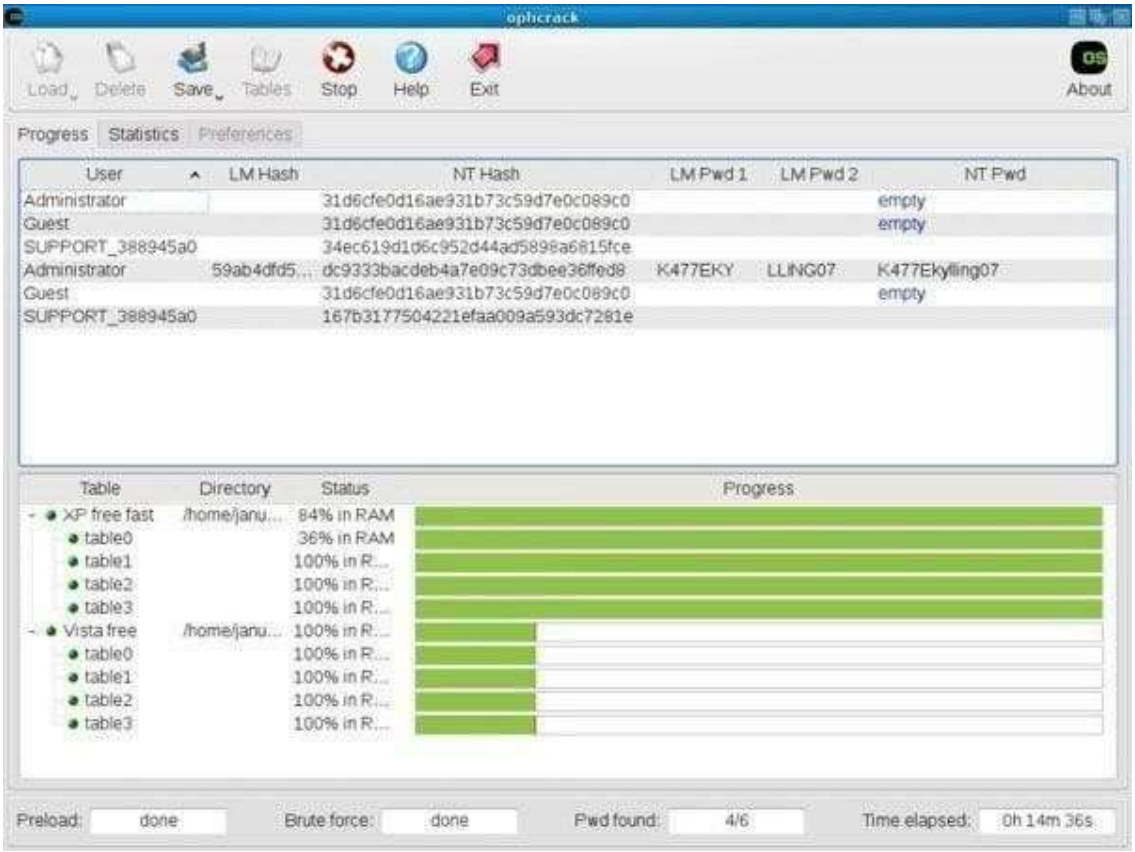


5. This will save the output to hashes.txt, which you can analyze or use with other tools (e.g., John the Ripper or Hashcat) for further security assessments or password-cracking tests.

C. Ophcrack

Ophcrack is a free, open-source password-cracking tool used primarily for recovering Windows passwords by using rainbow tables. Unlike brute-force attacks, which test passwords individually, Ophcrack relies on precomputed rainbow tables that map common password hashes to their plaintext equivalents. This approach can crack many common passwords much faster.

1. Load Password Hashes
2. Select Rainbow Tables
3. Run the Program



4. Review the Results

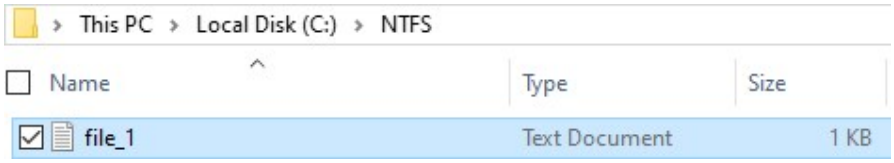
D. NTFS Stream Manipulation

NTFS stream manipulation refers to working with alternate data streams (ADS) in the NTFS file system. ADS are a unique feature of NTFS that allow files to contain additional hidden data streams, enabling a single file to store multiple data sets under a single file name. This feature can be useful for metadata storage but also poses security risks, as malicious software can use ADS to hide data and evade detection.

1. Open the terminal and type the following command to create a file named file_1.txt.

echo "This is file_1" > file_1.txt

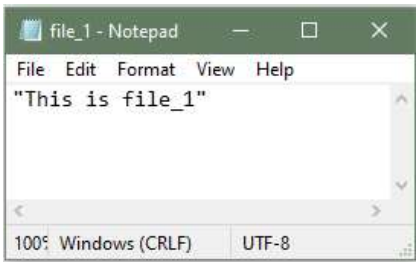
```
C:\NTFS>echo "This is file_1" > file_1.txt
```



2. Now, type the following command to write to the stream named secret.txt.
echo "This is a hidden file inside the file_1.txt" > file_1.txt:secret.txt

```
C:\NTFS>echo "This is a hidden file inside the file_1.txt" > file_1.txt:secret.txt
```

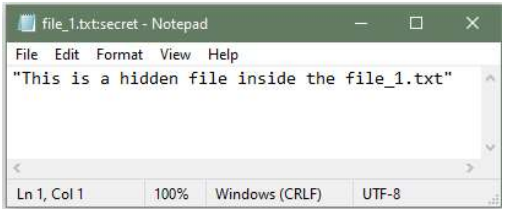
3. We've just created a stream named secret.txt that is associated with file_1.txt and when you look at the file_1.txt you will only find the data present in file_1.txt. And also stream will not be shown in the directory as well.




```
C:\NTFS>dir
 Volume in drive C has no label.
 Volume Serial Number is CE13-AB97

 Directory of C:\NTFS

23-11-2024  15:38    <DIR>          .
23-11-2024  15:38    <DIR>          ..
23-11-2024  15:44                19 file_1.txt
               1 File(s)             19 bytes
               2 Dir(s)  246,489,063,424 bytes free
```

4. The following command can be used to view or modify the stream hidden in file_1.txt.
notepad file_1.txt :secret.txt

```
C:\NTFS>notepad file_1.txt:secret.txt
```



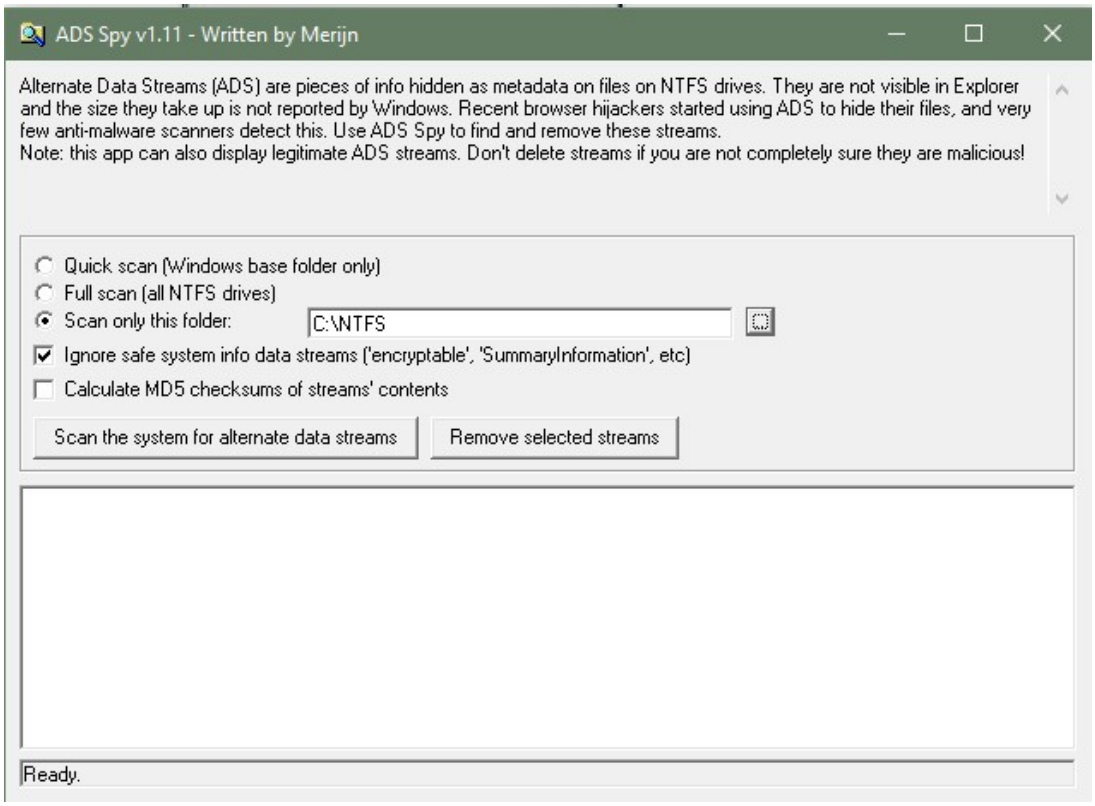
E. ADS Spy

ADS Spy offers the most search options of any Ad Intelligence Tool, so you can find the data you want, how you want. Search in the usual way: ad text, URL, page name. Search true data from user reactions in advert comments. Be as rigorous as you need to: search or filter by affiliate network, affiliate ID, Offer ID, landing page technologies - whatever helps you find the information you can work with.

Open ADS Spy application and select the option if you want to:

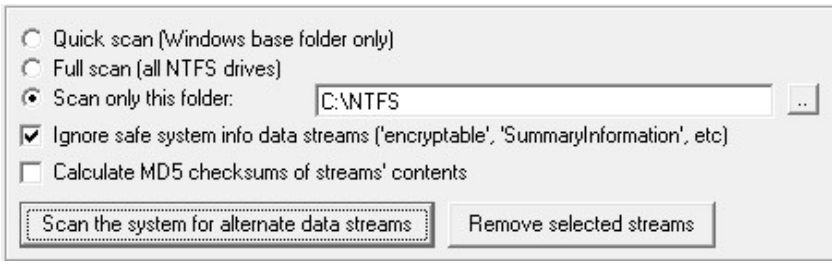
i. Quick Scan
ii. Full Scan
iii. Scan Specific Folder

1. As we store the file in the Document folder, Selecting Document folder to scan particular folder only.

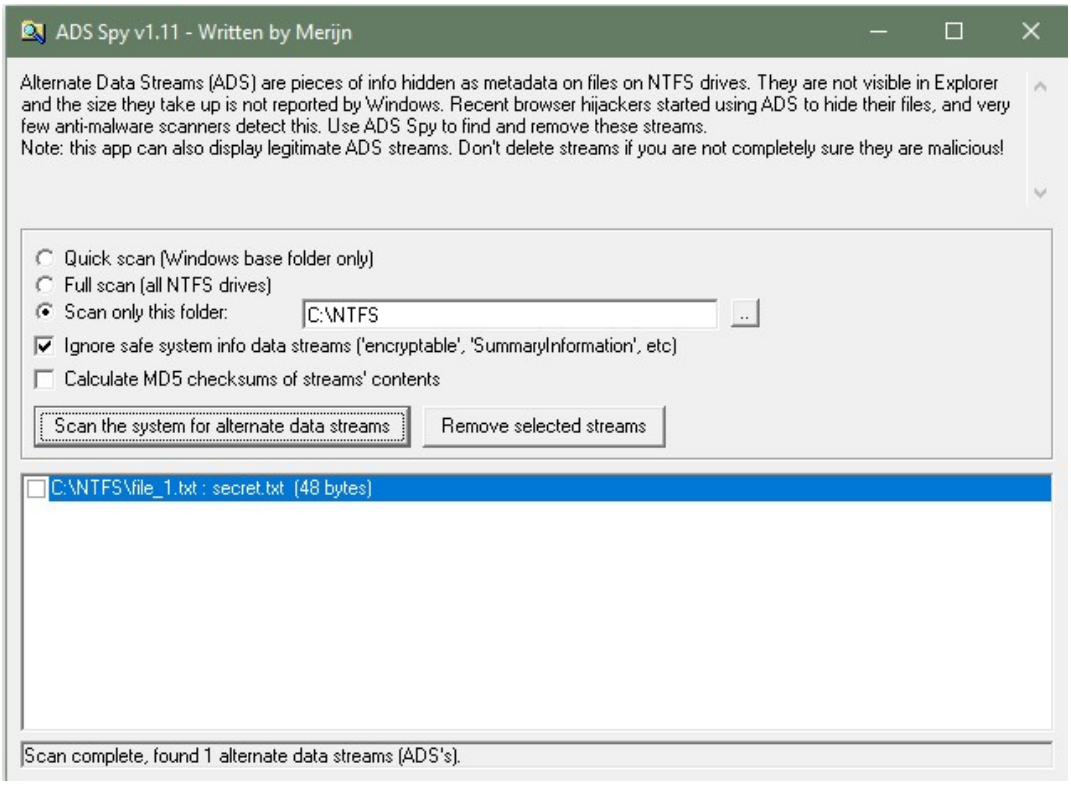


2. Select an Option, if you want to scan for ADS, click **Scan the system for ADS** or click

   Remove selected streams to remove the file



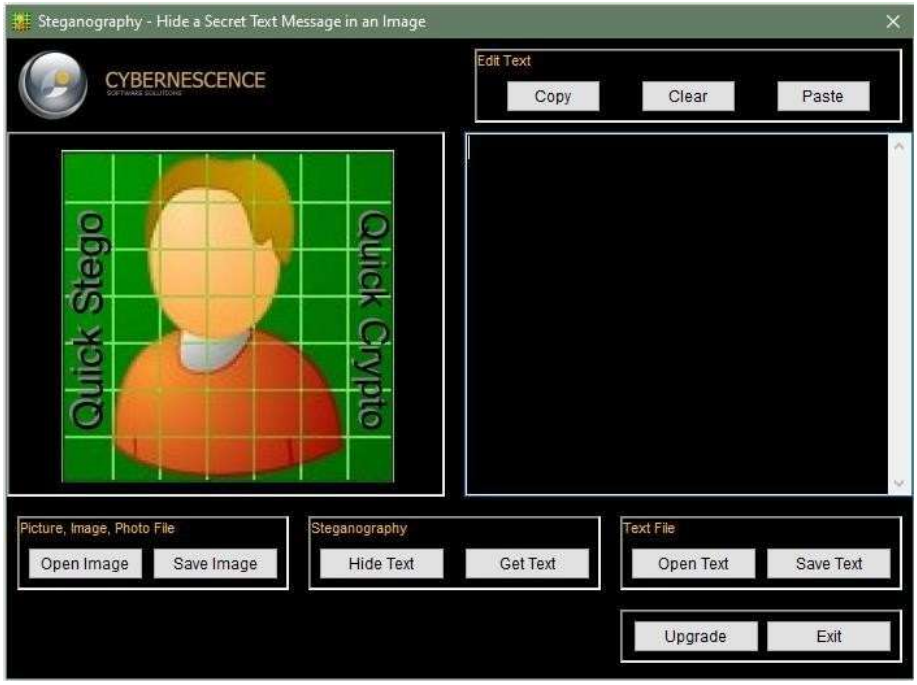
3. As shown in the figure below, ADS Spy has detected the **file_1.txt:secret.txt** file from the directory.



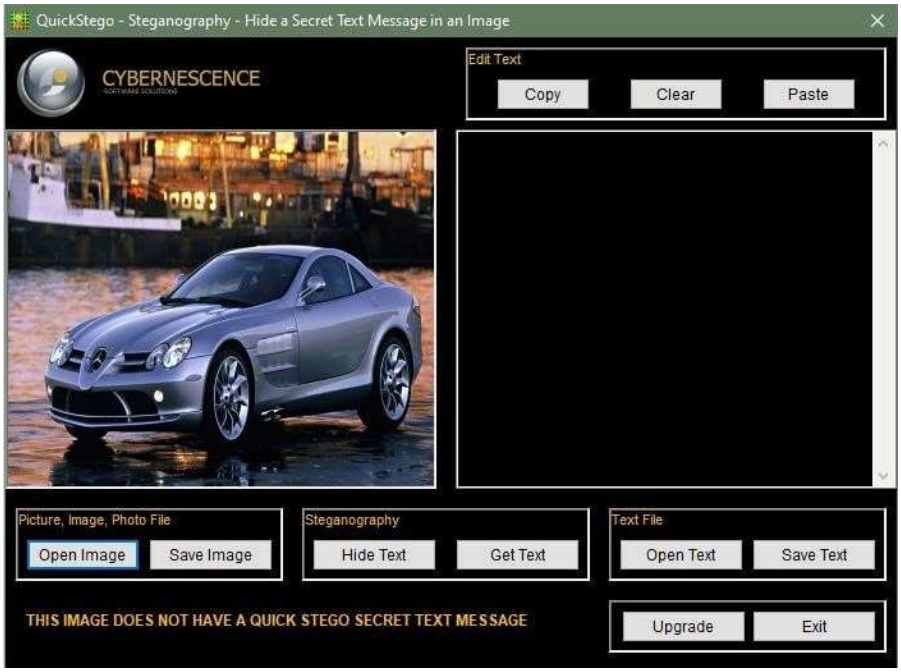
F. Quickstego

Quick Stego hides text in pictures so that only other users of Quick Stego can retrieve and read the hidden secret messages.
QuickStego website: http://quickcrypto.com/free-steganography-software.html

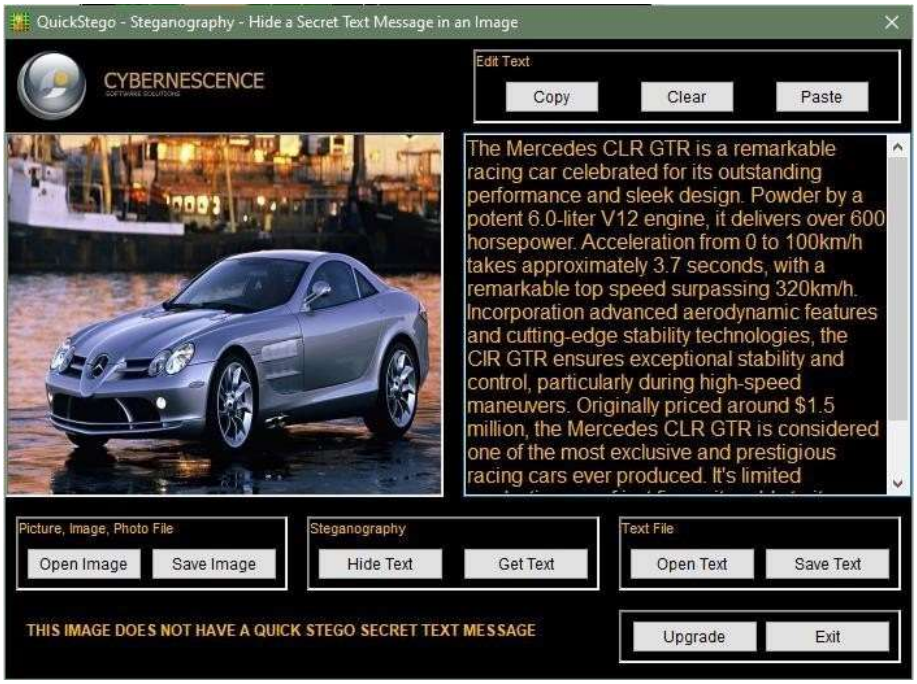
1. Open QuickStego Application



2. Upload an image. This image is term as **Cover**, as it will hide the text.

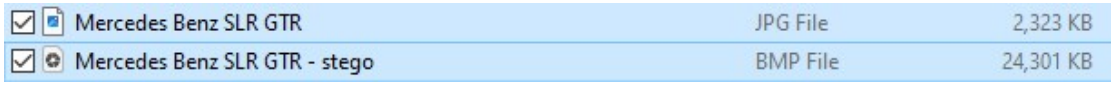


3. Enter the Text or Upload Text File.

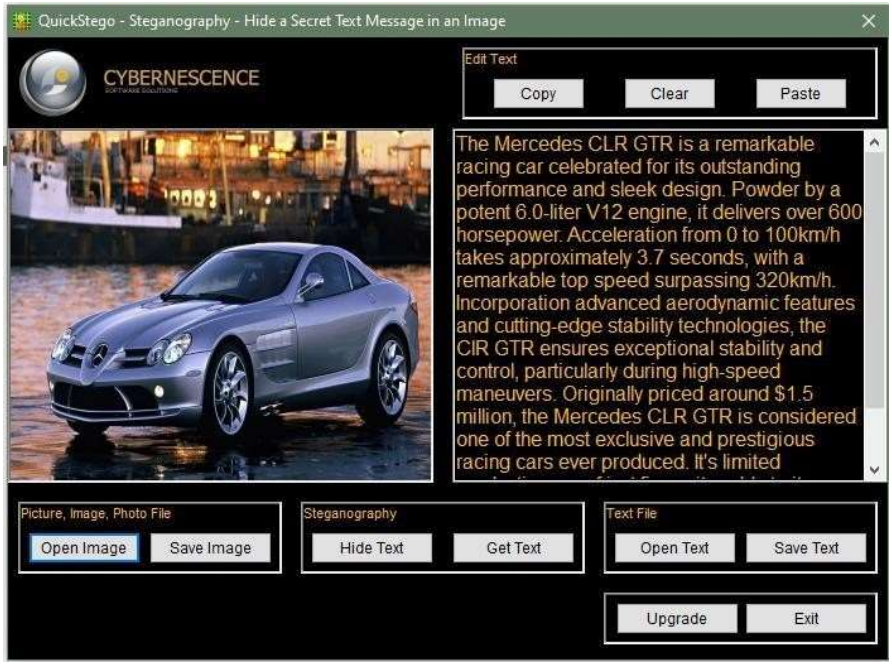


4. Click Hide Text Button

5. Save Image



6. To recover data from stego object, click on Get Text



Practical No. 6

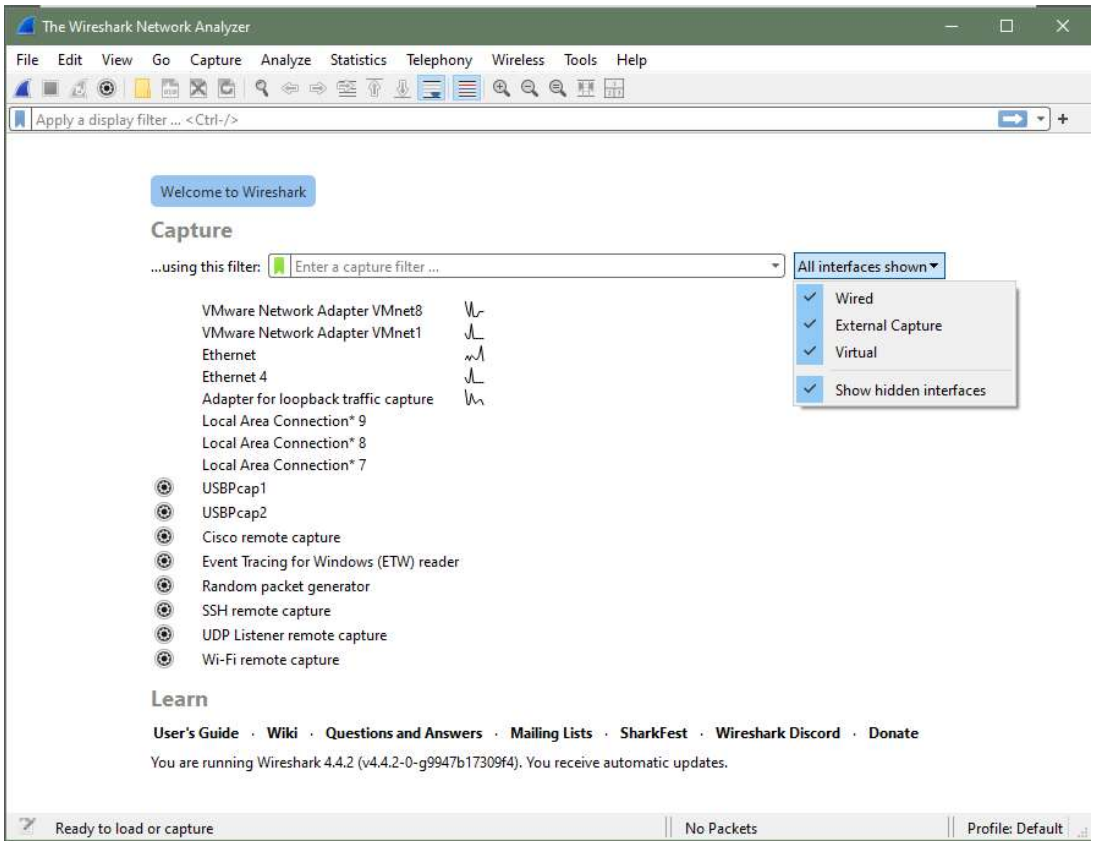
Aim:

A. **Sniff the network packet to break the password using Wireshark.**
B. **Change the MAC of the system using SMAC tool.**
C. **Perform the network analysis using Caspa Network Analyzer Tool.**

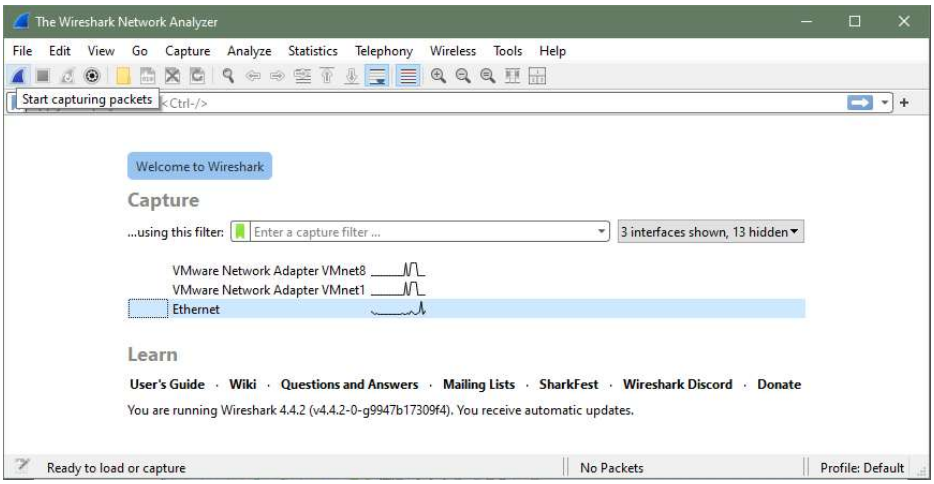
A. Sniff the network packet to break the password using Wireshark.

Wireshark is a powerful, open-source, GUI-based network protocol analyzer used by network administrators, security professionals, and developers to capture and examine network packets in real time. By analyzing network traffic, Wireshark can help with troubleshooting network issues, monitoring network security, and studying how protocols work.

1. Start Wireshark. Under the **Capture** header, select the **Interface List** option or click on the **Interfaces** button on the toolbar.
   This will bring up a list of network interfaces that Wireshark is able to capture packets from:

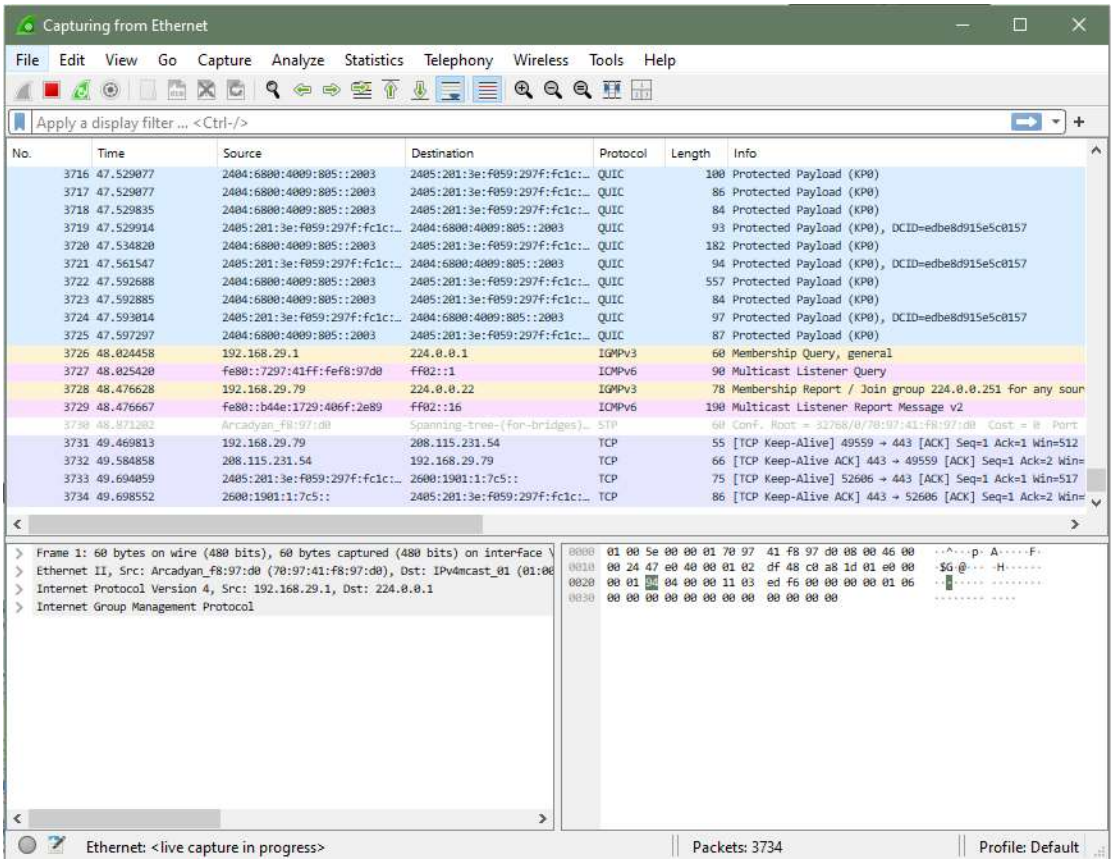


2. Select the network adapter (wired or wireless) that you are currently using to connect to the Internet, and hit the **Start** button. This will take you to the main window:

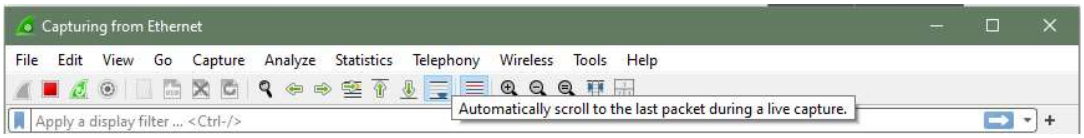


Wireshark is now capturing live network activity on your network interface. Notice that the list of packets is color-coded to highlight different types of network traffic.

3. Open your web browser and navigate to a few random web pages - observe that the network packets corresponding to your web browsing activity are captured and show up in Wireshark as well.

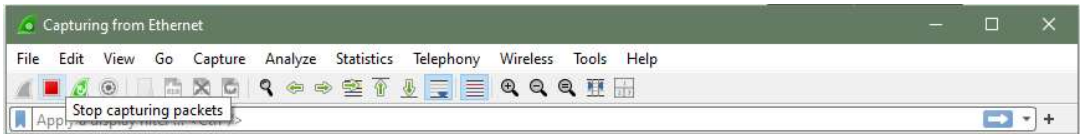


4. By default, the list of captured packets will keep scrolling automatically during a live capture. You can toggle this on/off using the AutoScroll toggle button in the toolbar.



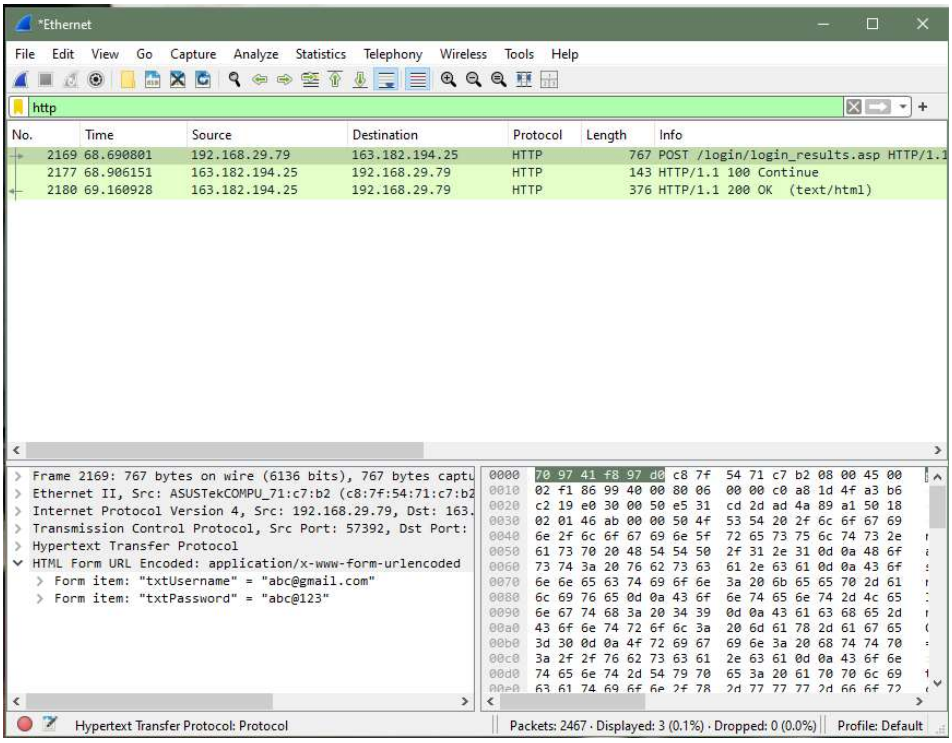
5. Visit a HTTP connection website and enter some login information. For example, http://vbsca.ca/login/login.asp Username: abc@gmail.com
Password: abc@123

6. After letting the capture run for a couple of minutes, press the stop capture button. Do not close this capture session.



Capturing network traffic for a couple minutes could include traffic on many different protocols such as ARP, TCP, UDP, DNS, HTTP, etc.
We may not be interested in all of these, depending on what we are trying to achieve. Fortunately, Wireshark allows us to filter the list based on different criteria using the "Filter" toolbar:

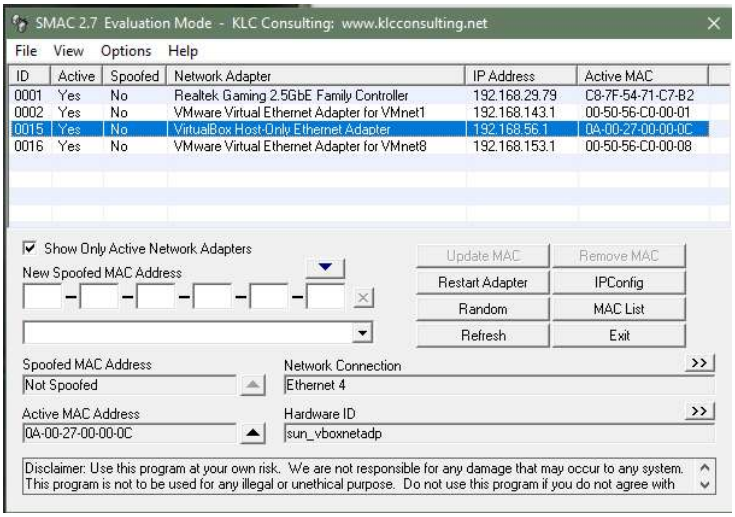
7. In the filter toolbar, type "http" and then click on "Apply". The window will now list only captured packets related to HTTP traffic:



B. Change the MAC of the system using SMAC tool.

SMAC is a Windows-based tool used to spoof, or change, the MAC (Media Access Control) address of a network adapter without changing the physical hardware. This allows users to bypass MAC-based security controls on networks, test network applications, and troubleshoot connectivity issues related to MAC address filtering. SMAC's user-friendly interface provides easy access to change the MAC address, view IP configurations, and reset network adapters after changes. It's commonly used by IT professionals for legitimate testing and security purposes but must be used responsibly and in compliance with network policies.

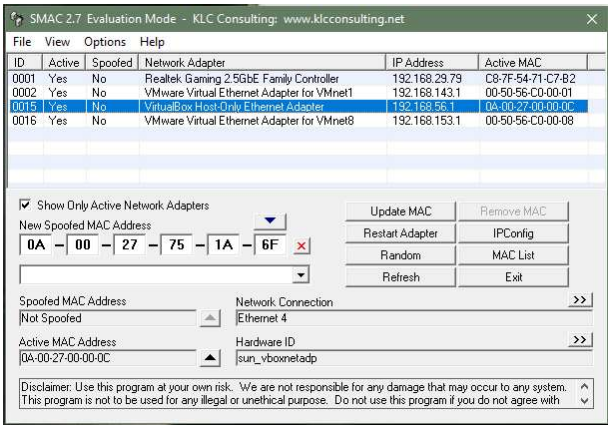
1. Open SMAC with administrative privileges to ensure it can interact with network adapters. Choose the desired network adapter from the list displayed in the application. This is the interface for which you want to change the MAC address.



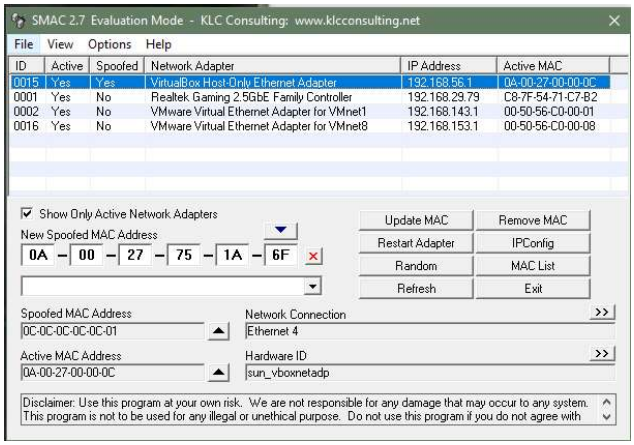
2. Click on **Random** to assign a random MAC address.

   OR

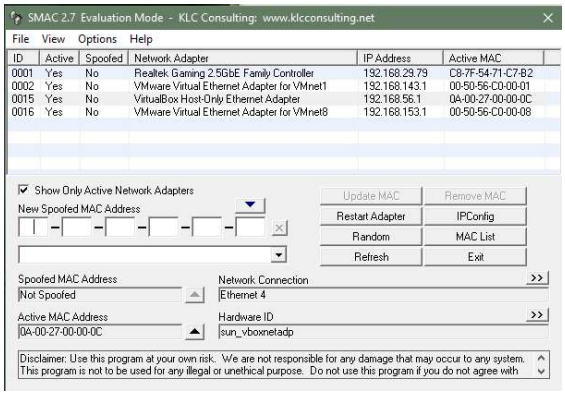
   Enter the desired MAC address in the provided field. Ensure it follows the standard hexadecimal format (e.g., 00-14-22-01-23-45).



3. Click the **Update MAC** or equivalent button to apply the new MAC address to the selected adapter.



4. If you want to restore the original MAC address, use the reset or restore feature within SMAC.

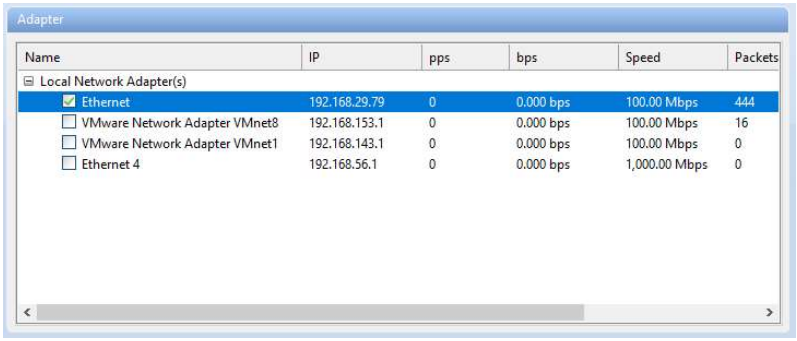


C. Perform the network analysis using Caspa Network Analyzer Tool.

Caspa Network Analyzer is a versatile tool designed for monitoring, capturing, and analyzing network traffic in real time. It supports a wide range of protocols and is useful for diagnosing network issues, identifying performance bottlenecks, and ensuring security compliance. With a user-friendly interface, Caspa provides packet-level inspection, allowing IT professionals to capture detailed traffic data across various network layers. This tool is particularly valuable for network administrators and security analysts who need in-depth visibility into their network's behavior, making it easier to detect anomalies or unauthorized access. Caspa also offers features like filtering, search functionality, and detailed logging for thorough analysis and reporting.

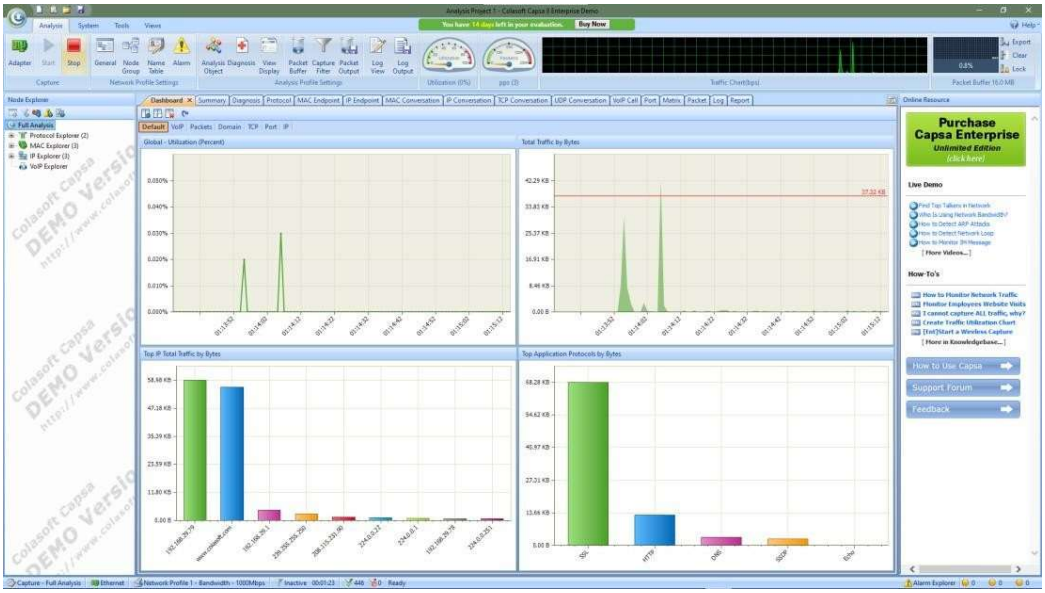
1. Open Caspa Network Analyzer Tool. In the Start Page, select your NICs (multiple selections available) in the Capture panel first.

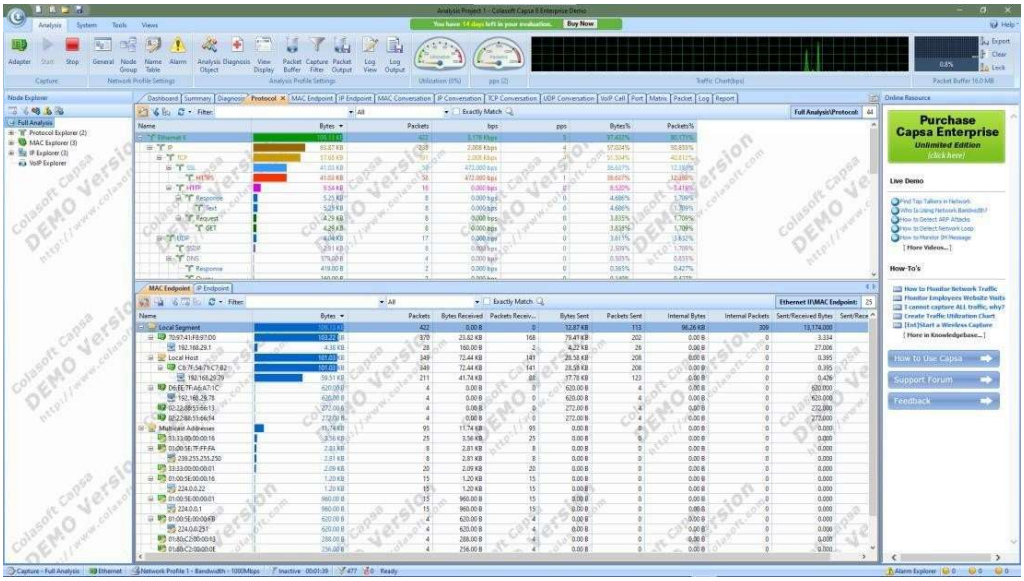
2. Select any Network Profile in the Network Profile panel.



3. Select Full Analysis in the Analysis Profile panel.

4. Click the big Run button to start a capture right away.





Practical No. 7

Aim:

A. Use the social engineering toolkit to perform social engineering attack.

B. Perform the DDoS Attack on a website using:

a. Golden Key

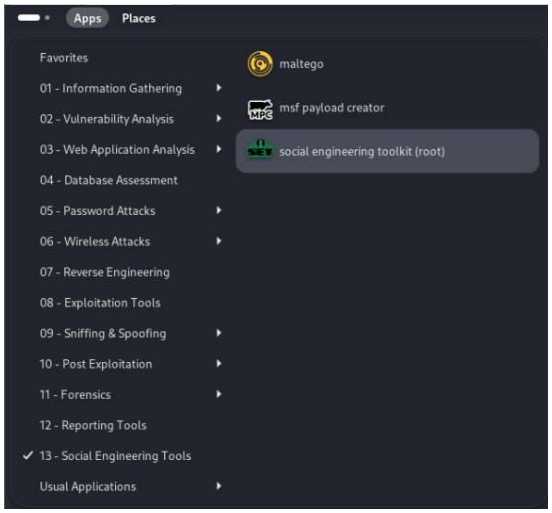
b. Metasploit

c. HOIC LOIC

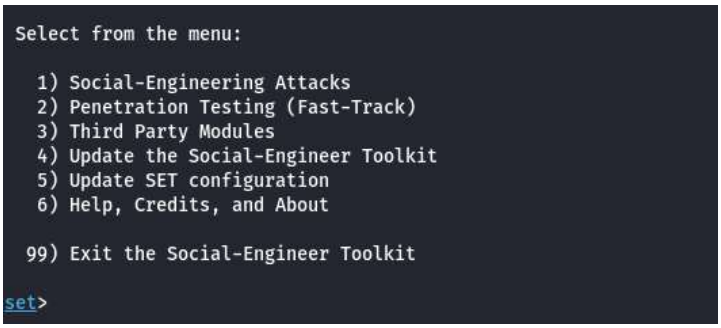
A. Use the social engineering toolkit to perform social engineering attack.

We are using Kali Linux Social Engineering Toolkit to clone a website and send clone link to victim. Once Victim attempt to login to the website using the link, his credentials will be extracted from Linux terminal.

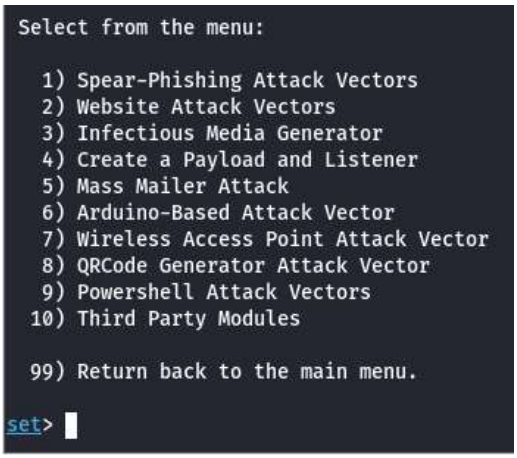
1. Open Kali Linux. Go to **Application □ Social Engineering Tools □ Social Engineering Toolkit**.



2. Type **1** for **Social Engineering Attacks**.

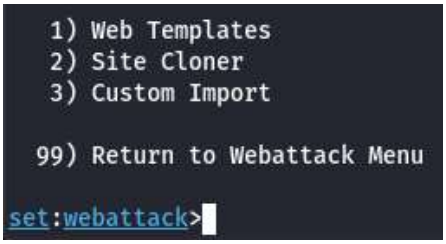


3. Type **2** for **Website Attack Vector**.



4. Type **3** for **Credentials Harvester Attack Method**.

5. Type **2** for **Site Cloner**.



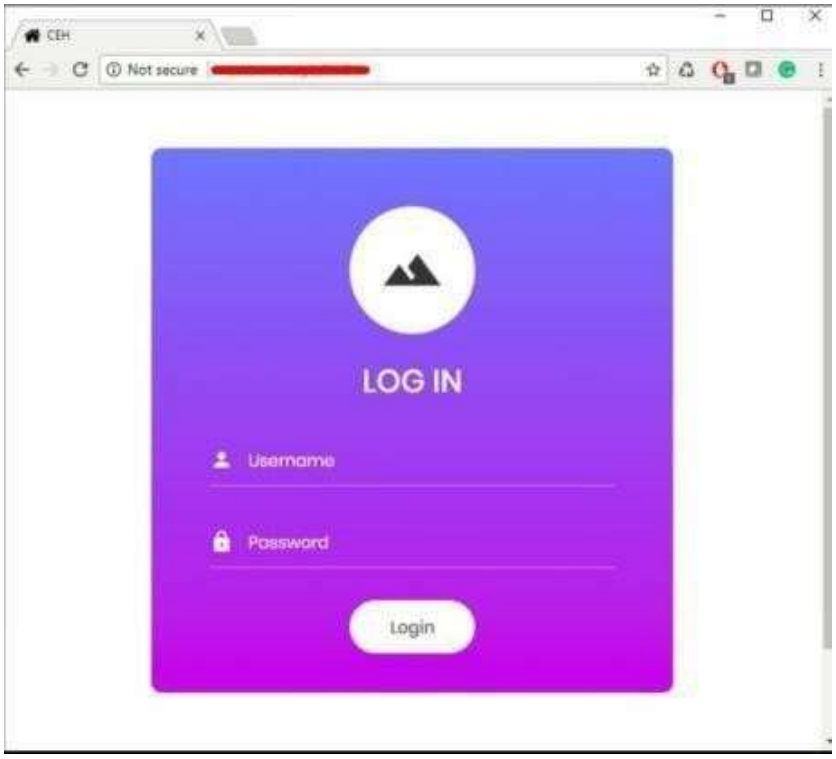
6. Type IP address of your Kali Linux machine. (192.168.153.128 in our case.)

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.153.128]: 192.168.153.128
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
```

7. Type Target URL.

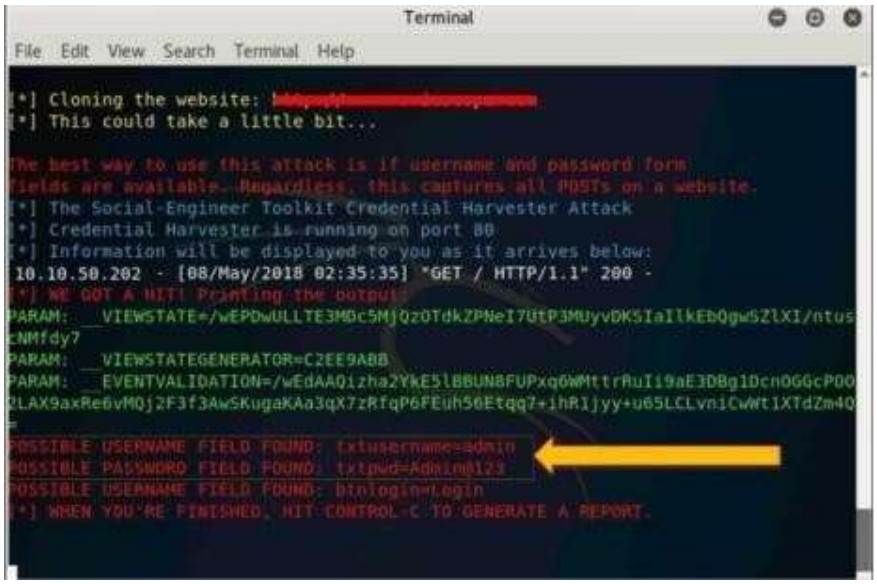
```
set:webattack> Enter the url to clone:

[*] Cloning the website: http://
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Regardless, this c
aptures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

8. Now, http://192.168.153.128 will be used. We can use this address directly, but it is not an effective way in real scenarios. This address is hidden in a fake URL and forwarded to the victim. Due to cloning, the user could not identify the fake website unless he observes the URL. If he accidentally clicks and attempts to log in, credentials will be fetched to Linux terminal. In the figure below, we are using http://192.168.153.128 to proceed.



9. Login using username and Password Username: admin
Password: Admin@123

10. Go back to Linux terminal and observe.



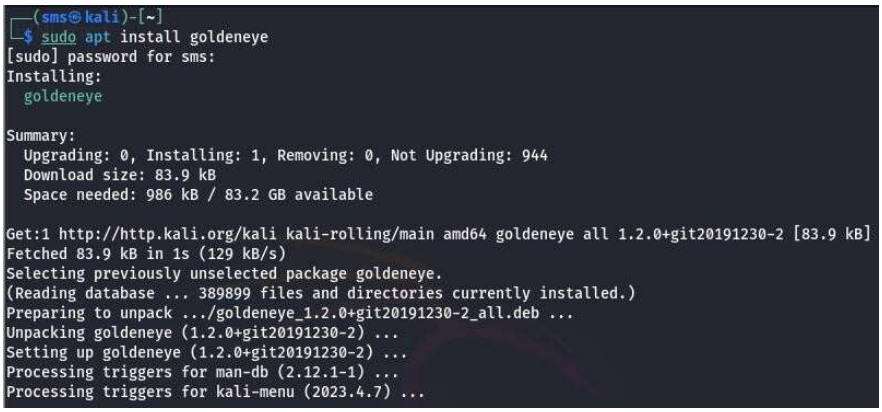
Username admin and password is extracted. If the user types it correctly, exact spelling can be used. However, you will get the closest guess of user ID and password. The victim will observe a page redirect, and he will be redirected to a legitimate site where he can re-attempt to log in and browse the site.

B. DDoS Attack

In a distributed denial-of-service attack (DDoS attack), the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source. A DDoS attack utilizes many sources of attack traffic, often in the form of a botnet. Generally speaking, many of the attacks are fundamentally similar and can be attempted using one more many sources of malicious traffic.

a. Golden Eye

1. Install the package using command : **sudo apt install goldeneye**



2. Type **goldeneye** to check whether it is installed.

```
┌──(sms㉿kali)-[~]
└─$ goldeneye
Please supply at least the URL

------------------------------------------------------------------------------------------------------------

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

 USAGE: goldeneye <url> [OPTIONS]

 OPTIONS:
        Flag                 Description                                          Default
        -u, --useragents     File with user-agents to use                         (default: randomly generated)
        -w, --workers        Number of concurrent workers                         (default: 10)
        -s, --sockets        Number of concurrent sockets                         (default: 500)
        -m, --method         HTTP Method to use 'get' or 'post'  or 'random'      (default: get)
        -n, --nosslcheck     Do not verify SSL Certificate                        (default: True)
        -d, --debug          Enable Debug Mode [more verbose output]              (default: False)
        -h, --help           Shows this help

------------------------------------------------------------------------------------------------------------
```

3. Perform a DDoS attack by typing the following command : **goldeneye *<target URL>***

```
┌──(sms㉿kali)-[~]
└─$ goldeneye http://www.certifiedhacker.com

GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>

Hitting webserver in mode 'get' with 10 workers running 500 connections each. Hit CTRL+C to cancel.
0 GoldenEye strikes hit. (2060 Failed)
0 GoldenEye strikes hit. (3244 Failed)
0 GoldenEye strikes hit. (4247 Failed)
0 GoldenEye strikes hit. (5113 Failed)
```

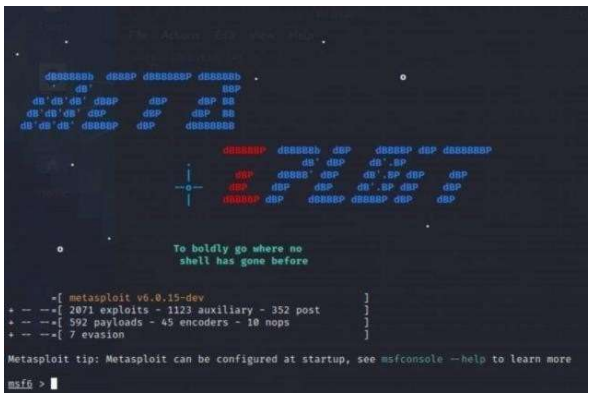
b. Metasploit

First, select your target's IP address. I am taking testphp.vulnweb.com as a victim. So you know how to get an IP address from a domain name. Simple doping and that will give to domain IP address.

1. So now I know the victim's IP Address 18.192.182.30.

```
┌──(kali㉿kali)-[~]
└─$ ping testphp.vulnweb.com
PING testphp.vulnweb.com (18.192.172.30) 56(84) bytes of data.
64 bytes from ec2-18-192-172-30.eu-central-1.compute.amazonaws.com (18.192.
172.30): icmp_seq=1 ttl=39 time=206 ms
64 bytes from ec2-18-192-172-30.eu-central-1.compute.amazonaws.com (18.192.
172.30): icmp_seq=2 ttl=39 time=228 ms
^C
--- testphp.vulnweb.com ping statistics ---
3 packets transmitted, 2 received, 33.3333% packet loss, time 2004ms
rtt min/avg/max/mdev = 205.509/216.576/227.643/11.067 ms
```

2. Launching Metasploit by typing msfconsole in your kali terminal.

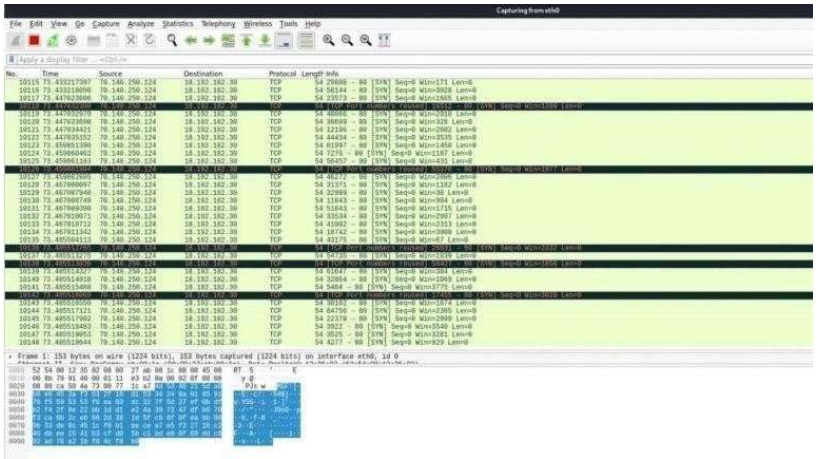


3. Then use the select the auxiliary “auxiliary/dos/TCP/synflood” by typing the following command.
   Msf6 > use auxiliary/dos/tcp/synflood
   Msf6> show options

4. Now you can see you have all the available options that you can set.
   To set an option just you have to typeset and the option name and option.
   You have to set two main option
   RHOST= target IP Address
   RPORT=target PORT Address
   Set RPORT 18.192.182.30
   Set RPORT 80

5. To launch the attack just type: **exploit**

6. To see the packets you can open Wireshark.

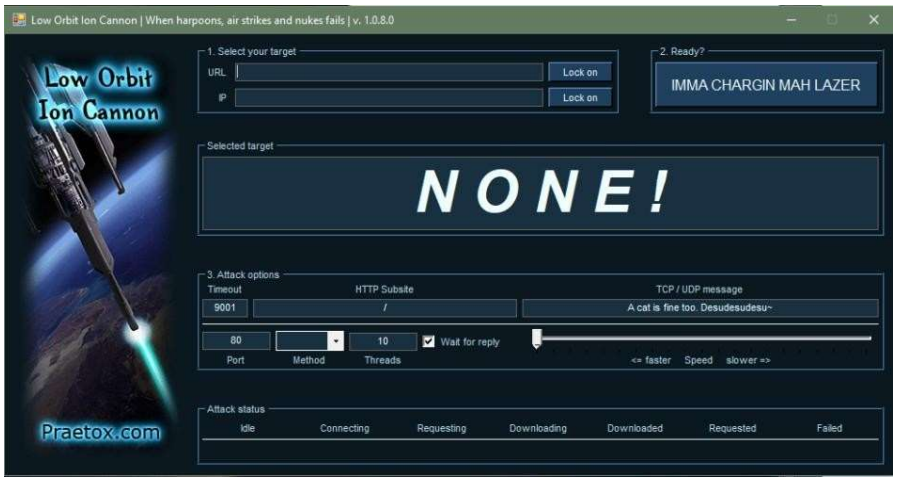


c. HOIC LOIC

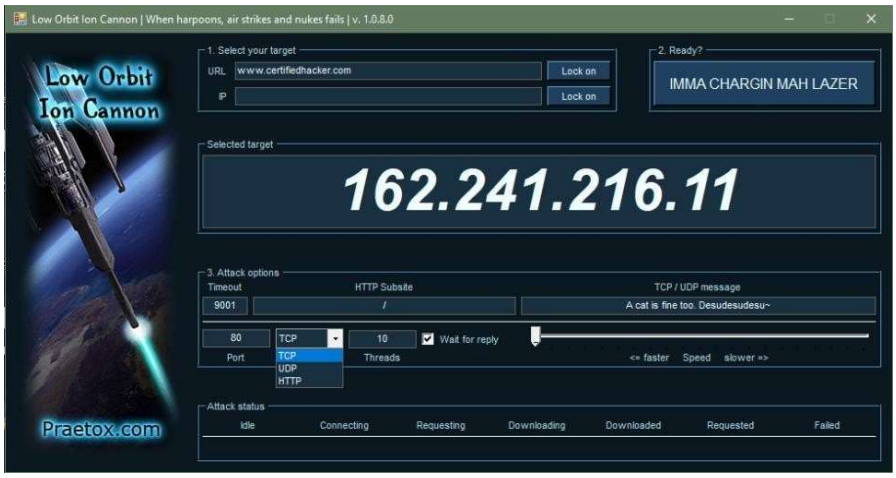
The **Low Orbit Ion Cannon (LOIC)** was originally developed by Praetox Technologies as a stress testing application before becoming available within the public domain. The tool is able to perform a simple dos attack by sending a large sequence of UDP, TCP or HTTP requests to the target server. It's a very easy tool to use, even by those lacking any basic knowledge of hacking. The only thing a user needs to know for using the tool is the URL of the target. A would-be hacker need only then select some easy options (address of target system and method of attack) and click a button to start the attack.

The tool takes the URL of the target server on which you want to perform the attack. You can also enter the IP address of the target system. The IP address of the target is used in place of an internal local network where DNS is not being used. The tool has three chief methods of attack: TCP, UDP and HTTP. You can select the method of attack on the target server. Some other options include timeout, TCP/UDP message, Port and threads. See the basic screen of the tool in the snapshot above in Figure.

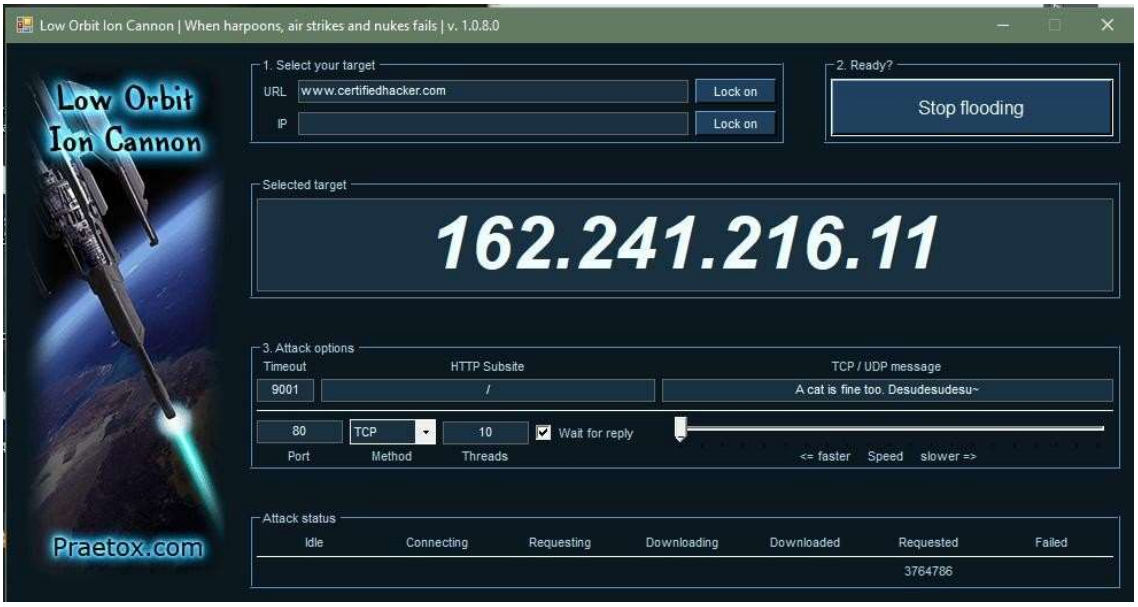
1. Open Low Orbit Ion Cannon (LOIC) tool.



2. Enter the URL of the website in The **URL field** and click on **Lock On**. Then, select attack method (TCP, UDP or HTTP). I will recommend TCP to start. These 2 options are necessary to start the attack.



3. Change other parameters per your choice or leave it to the default. Now click on the button labeled as **IMMA CHARGIN MAH LAZER**. You have just mounted an attack on the target.



After starting the attack you will see some numbers in the Attack status fields. When the requested number stops increasing, restart the LOIC or change the IP. You can also give the UDP attack a try. Users can also set the speed of the attack by the slider. It is set to faster as default but you can slow down it with the slider. I don't think anyone is going to slow down the attack.

The **High Orbit Ion Cannon (HOIC)** is a free, open-source network stress application developed by Anonymous, a hacktivist collective, to replace the Low Orbit Ion Cannon (LOIC). Used for denial of service (DoS) and distributed denial of service (DDoS) attacks, it functions by flooding target systems with junk HTTP GET and POST requests.
Widespread HOIC availability means that users having limited knowledge and experience can execute potentially significant DDoS attacks. The application can open up to 256 simultaneous attack sessions at once, bringing down a target system by sending a continuous stream of junk traffic until legitimate requests are no longer able to be processed.

Practical No. 8

Aim:

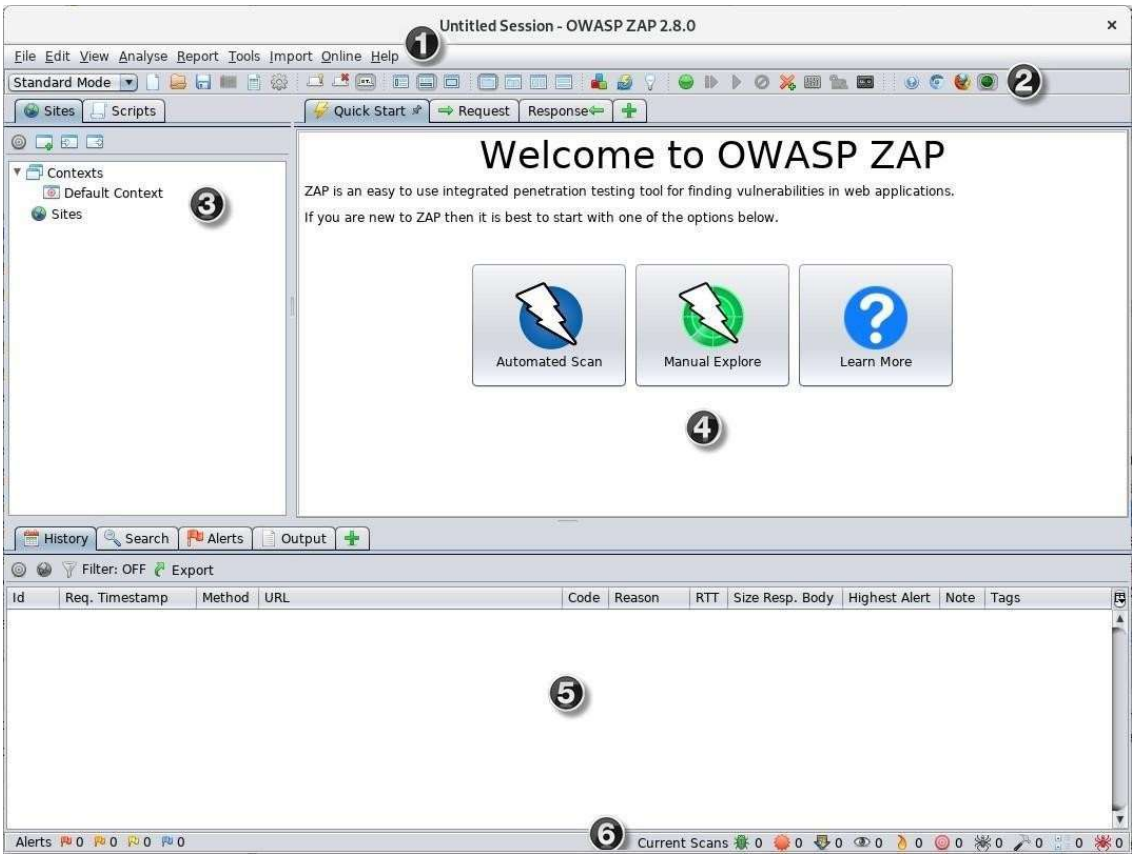
A. **Perform the Web Scanning using OWSAP Zed Proxy.**
B. **Use the HoneyBOT to capture malicious network traffic.**

A. Perform the Web Scanning using OWSAP Zed Proxy.

Zed Attack Proxy (ZAP) is a free, open-source penetration testing tool being maintained under the umbrella of the Open Web Application Security Project (OWASP). ZAP is designed specifically for testing web applications and is both flexible and extensible.
At its core, ZAP is what is known as a "man-in-the-middle proxy." It stands between the tester's browser and the web application so that it can intercept and inspect messages sent between browser and web application, modify the contents if needed, and then forward those packets on to the destination. It can be used as a stand-alone application, and as a daemon process.

To run a Quick Start Automated Scan:

1. Start ZAP and click the **Quick Start** tab of the Workspace Window.



2. Click the **Automated Scan button**.

3. In the **URL to Attack** text box, enter the full URL of the web application you want to attack.

4. Click the **Attack**

ZAP will proceed to crawl the web application with its spider and passively scan each page it finds. Then ZAP will use the active scanner to attack all of the discovered pages, functionality, and parameters.

ZAP provides 2 spiders for crawling web applications, you can use either or both of them from this screen.

The traditional ZAP spider which discovers links by examining the HTML in responses from the web application. This spider is fast, but it is not always effective when exploring an AJAX web application that generates links using JavaScript.

B. Use the HoneyBOT to capture malicious network traffic.

HoneyBot is a set of scripts and libraries for capturing and analyzing packet captures with PacketTotal.com. Currently, this library provides three scripts:

- `capture-and-analyze.py` - Capture on an interface for some period of time, and upload capture for analysis.
- `upload-and-analyze.py` - Upload and analyze multiple packets captures to PacketTotal.com.
- `trigger-and-analyze.py` - Listen for unknown connections, and begin capturing when one is made. Captures are automatically uploaded and analyzed.

capture-and-analyze.py

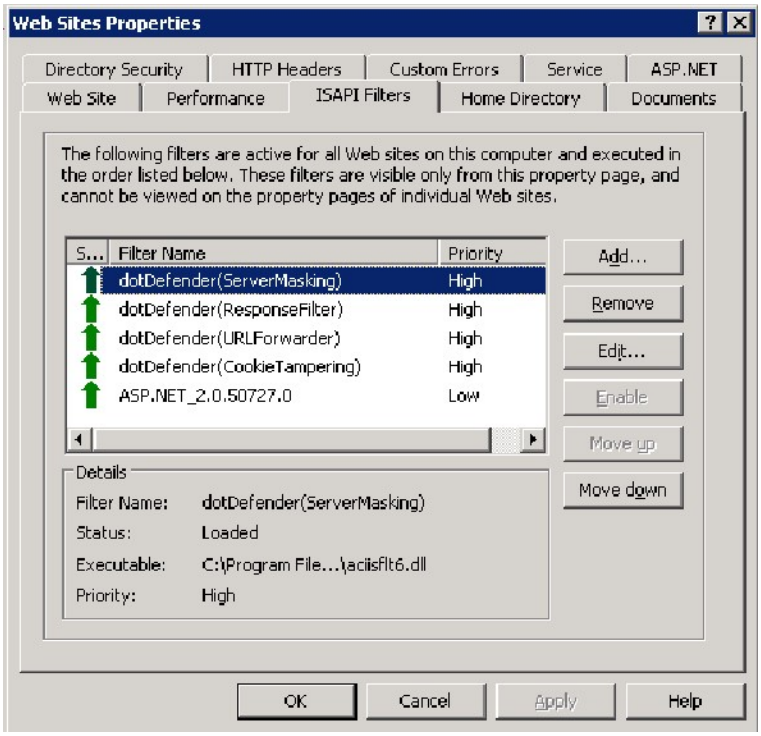
```
usage: capture-and-analyze.py [-h] [--seconds SECONDS] [--interface
INTERFACE]
                              [--analyze] [--list-interfaces] [--list-pcaps]
                              [--export-pcaps]

Capture, upload and analyze network traffic; powered by PacketTotal.com.

optional arguments:
  -h, --help            show this help message and exit
  --seconds SECONDS     The number of seconds to capture traffic for.
  --interface INTERFACE
                        The name of the interface (--list-interfaces to show
                        available)
  --analyze             If included, capture will be uploaded for analysis to
                        PacketTotal.com.
  --list-interfaces     Lists the available interfaces.
  --list-pcaps          Lists pcaps submitted to PacketTotal.com for
analysis.
  --export-pcaps        Writes pcaps submitted to PacketTotal.com for
analysis
                        to a csv file.
```

Practical No. 9

A. **Aim: A. Use Proxy workbentch Tool**
B. Perform Network Discovery using following tools:
a.LANState Pro
b.Network View
c.OpManager
Protect the web application using dotDefender.
Perform the database attack using SQL Injection Technique.

A. Protect the web application using dotDefender.

dotDefender allows businesses to protect external websites and internal applications in an affordable, effective and simple manner without involving costly security experts. dotDefender is a multi-platform solution running on Apache and IIS web servers. Central management ensures a single point of control and reporting for all servers.

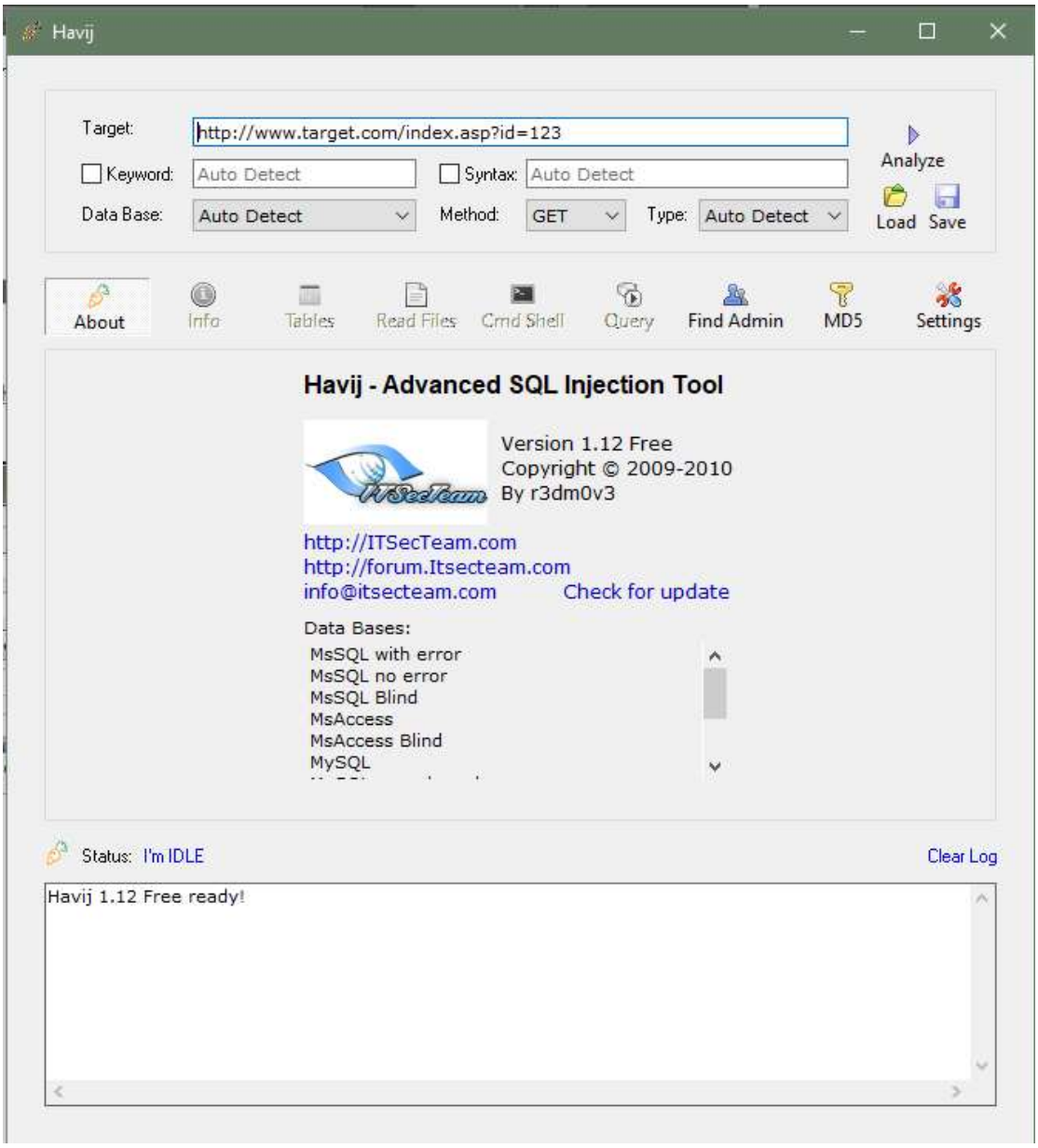


You can modify the Default Security Profile or any of the Website Security Profiles.

B. Perform the database attack using SQL Injection Technique.

a. Havij

Havij is an automated SQL Injection tool that helps penetration testers to find and exploit SQL Injection vulnerabilities on a web page.



Practical No. 10

Aim: Use the following cryptography tool to encrypt and decrypt the messages:

A. **HashCalc**
B. **CrypTool**
C. **TrueCrypt**

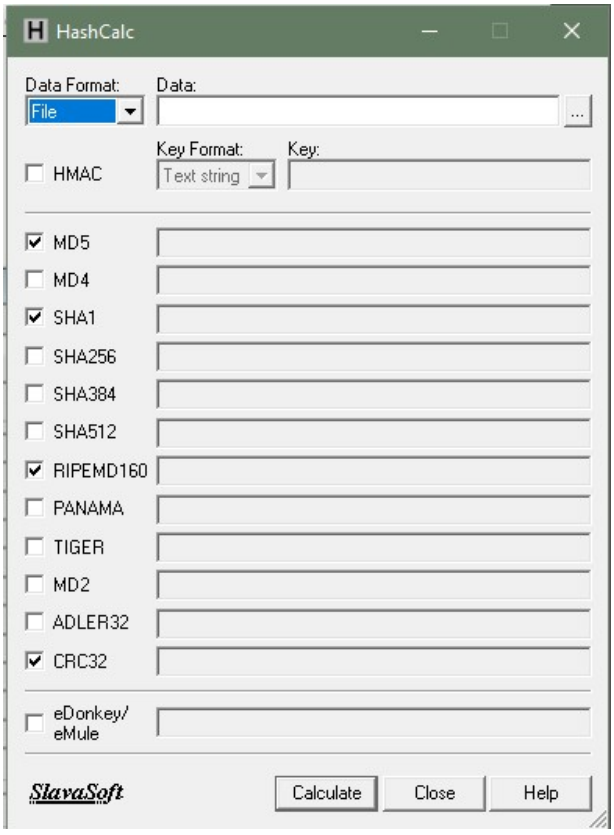
Crptography

Cryptography is the science of securing information by transforming it into an unreadable format, ensuring that only authorized parties can access it. This transformation is achieved through encryption algorithms, which encode data (plaintext) into a scrambled form (ciphertext) and can only be deciphered back with a decryption key. There are two main types of cryptography: symmetric-key, where the same key is used for both encryption and decryption, and asymmetric-key, which uses a pair of keys—a public key for encryption and a private key for decryption.

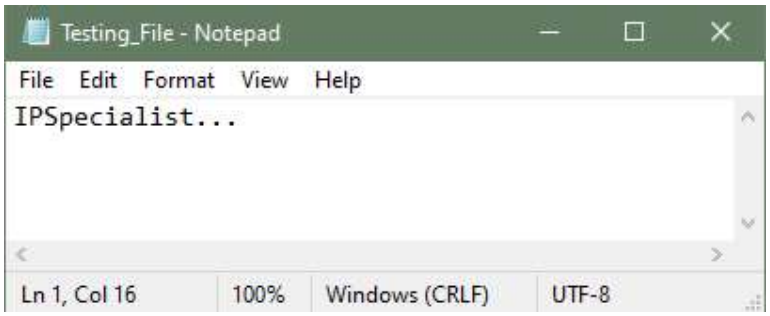
A. HashCalc

HashCalc is a free tool used for calculating cryptographic hash values for files or text. It supports several popular hash algorithms, such as MD5, SHA-1, SHA-256, and CRC32, providing users with the ability to verify file integrity or generate unique file fingerprints. The tool allows users to compute and compare checksums, making it useful for verifying downloaded files or ensuring data consistency. HashCalc also supports the calculation of hash values for large files, helping users check whether a file has been altered.

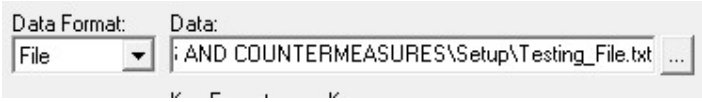
1. Open HashCalc tool.



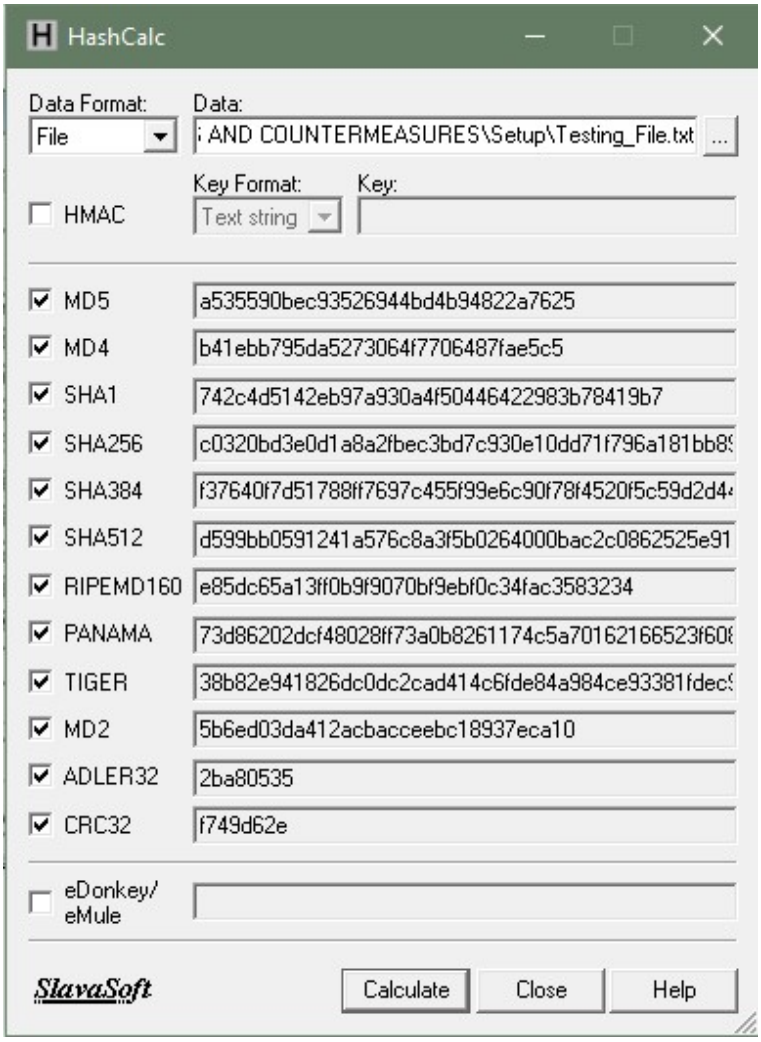
2. Create a new file with some content in it as shown below.



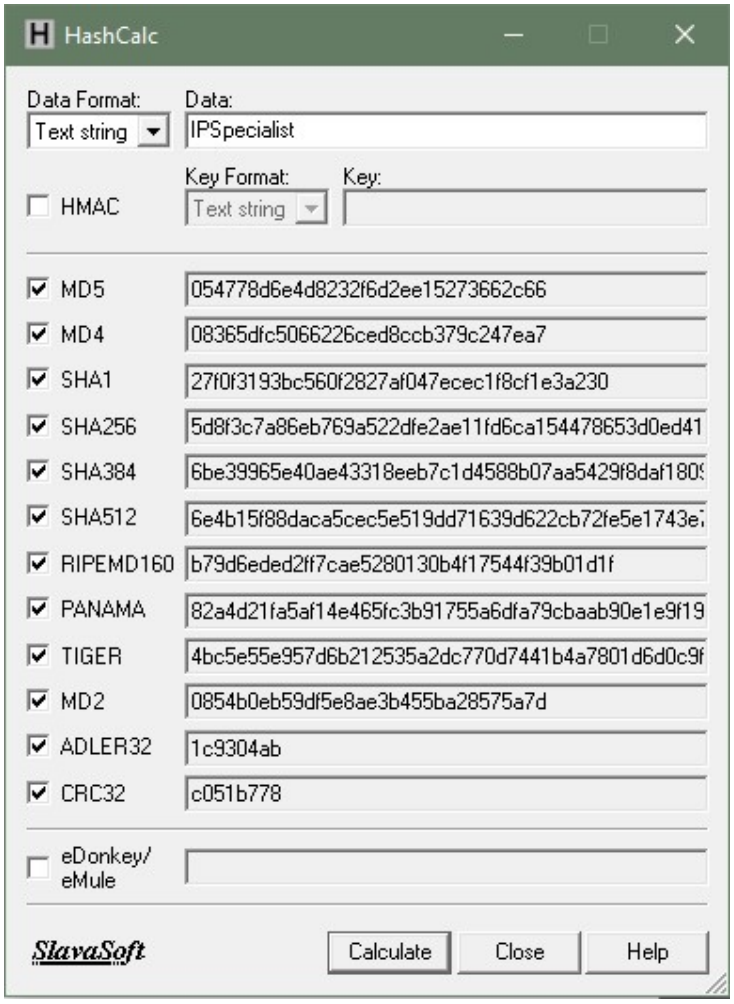
3. Select Data Format as “File” and upload your file.



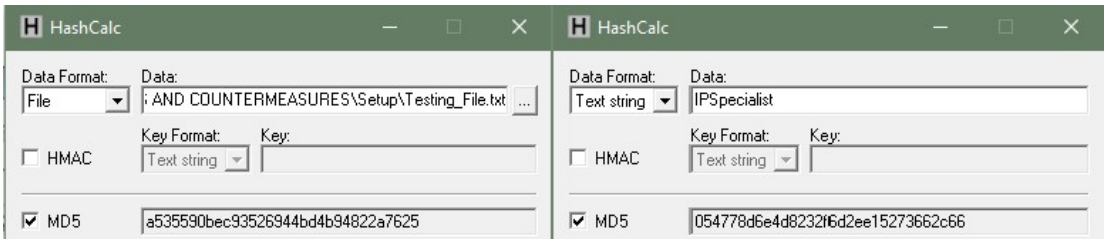
4. Select Hashing Algorithm and Click Calculate



5. Now Select the Data Format to “Text String” and Type “IPSpecialist” into Data filed and calculated MD5.

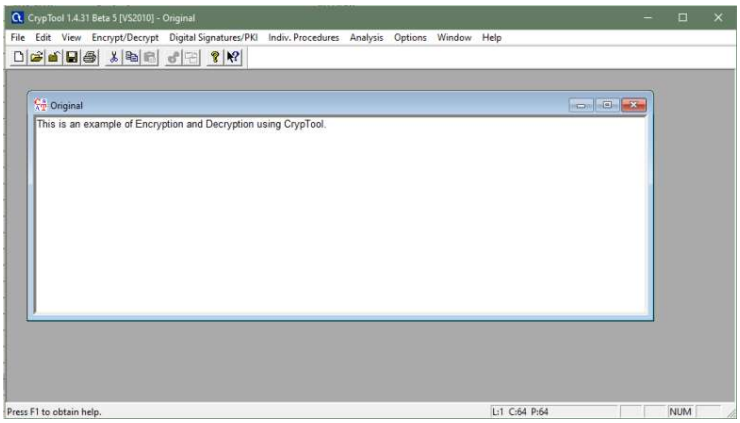


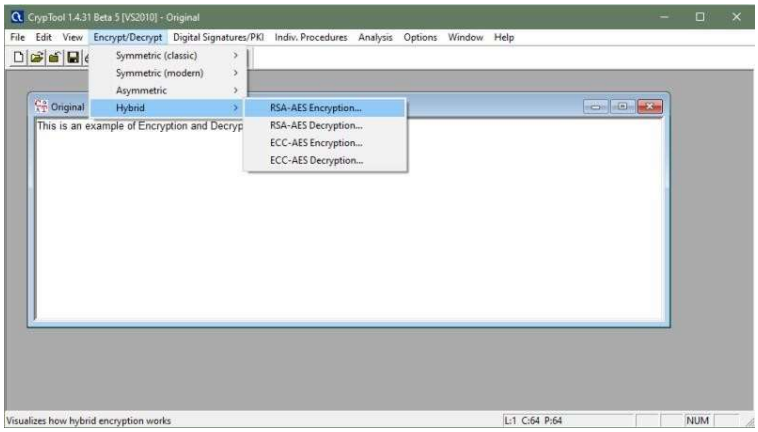
6. Now, let's see how MD5 value has a minor change.

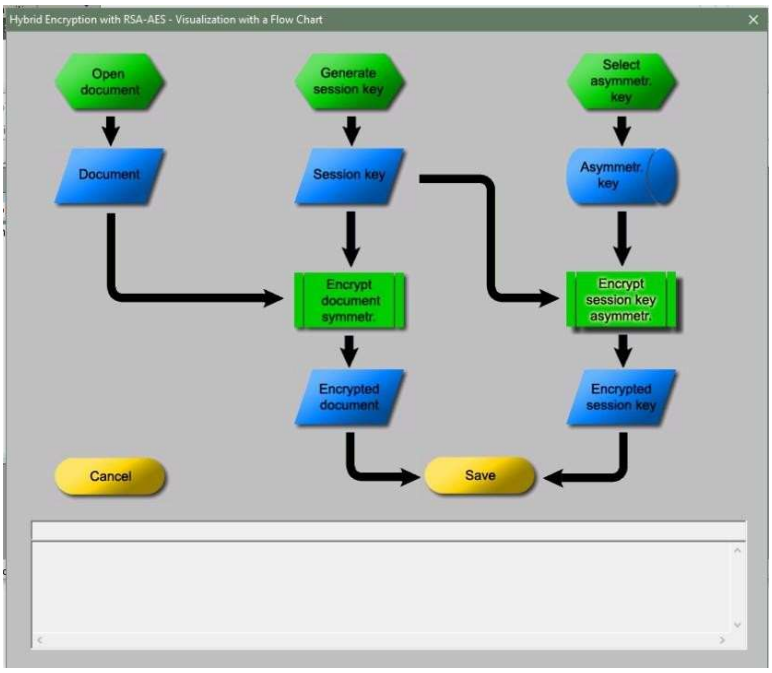


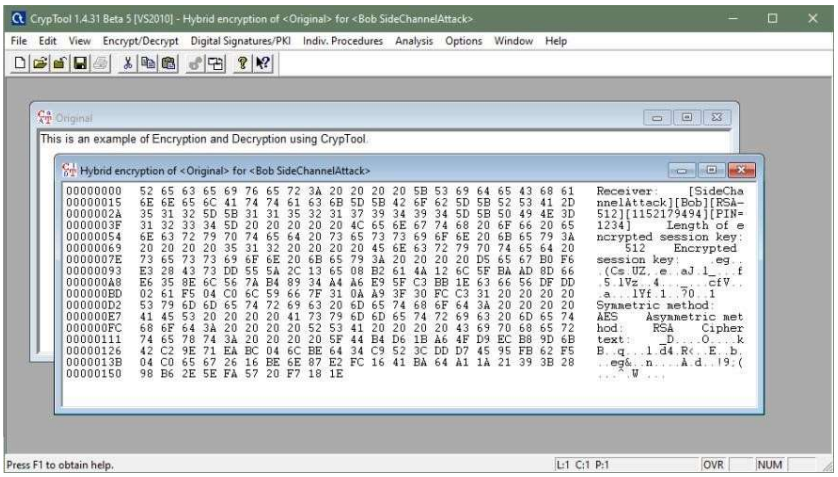
B. CrypTool

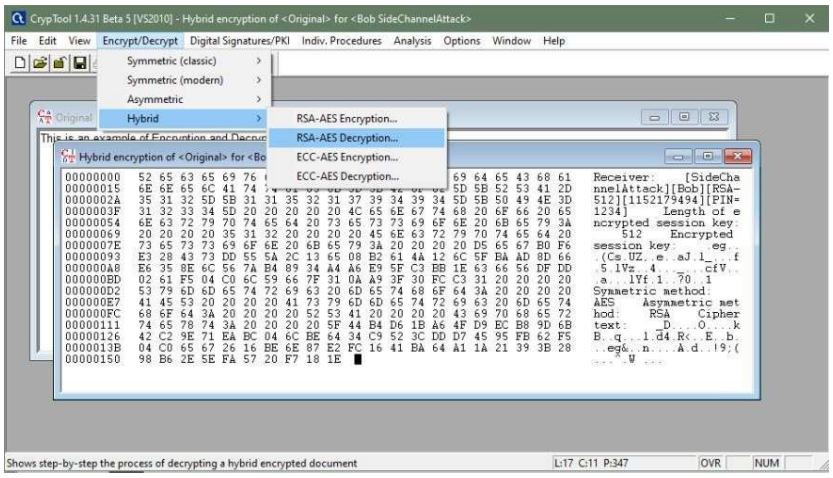
Cryptool is a free e-learning tool to illustrate the concepts of cryptography. Try Various Encryption/Decryption algorithms.

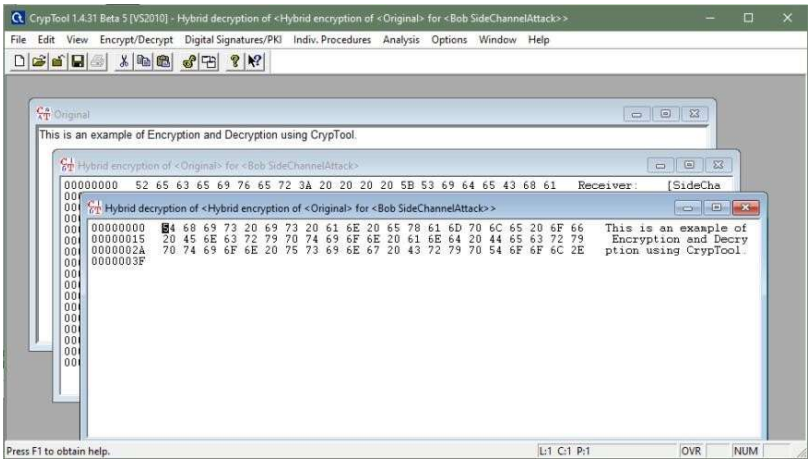






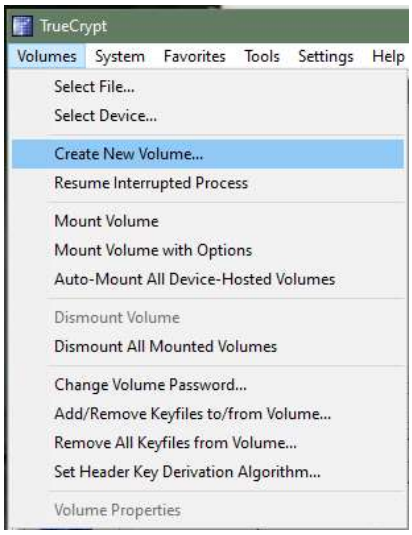

CrypTool 1.4.31 Beta 5 [VS2010] - Hybrid encryption of <Original> for <Bob SideChannelAttack>
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Original
This is an example of Encryption and Decryption using CrypTool.
Hybrid encryption of <Original> for <Bob SideChannelAttack>
Press F1 to obtain help.
L:1 C:1 P:1
OVR
NUM


CrypTool 1.4.31 Beta 5 [VS2010] - Hybrid encryption of <Original> for <Bob SideChannelAttack>
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Symmetric (classic)
Symmetric (modern)
Asymmetric
Hybrid
RSA-AES Encryption...
RSA-AES Decryption...
ECC-AES Encryption...
ECC-AES Decryption...
Shows step-by-step the process of decrypting a hybrid encrypted document
L:17 C:11 P:347
OVR
NUM

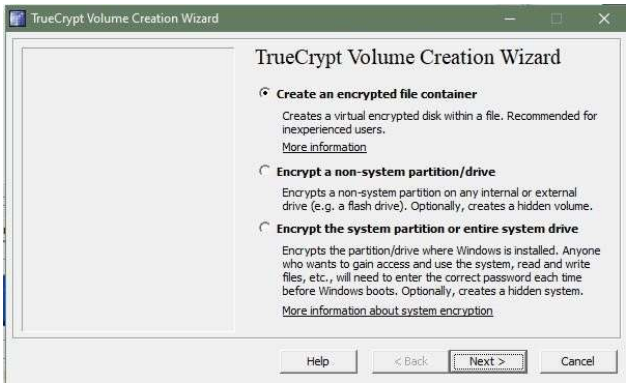

CrypTool 1.4.31 Beta 5 [VS2010] - Hybrid decryption of <Hybrid encryption of <Original> for <Bob SideChannelAttack>>
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
Original
This is an example of Encryption and Decryption using CrypTool.
Hybrid encryption of <Original> for <Bob SideChannelAttack>
Hybrid decryption of <Hybrid encryption of <Original> for <Bob SideChannelAttack>>
Press F1 to obtain help.
L:1 C:1 P:1
OVR
NUM

C. TrueCrypt

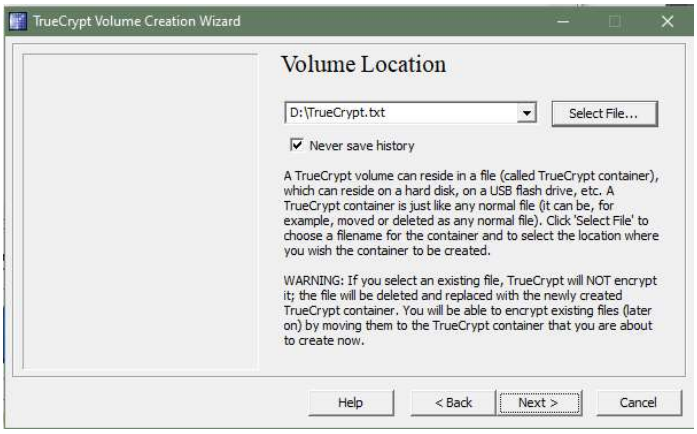
TrueCrypt is a leading disk encryption software program that lets you secure disk partitions on your Windows computer. There are times when your hard drive is accessible by other people, such as in an office setting, while travelling, or at home. The data you have on the PC may be vulnerable to attack and compromise your privacy. However, in these moments of risk, TrueCrypt may just be the tool to protect your data.



1. Click Next two times on the following screens to create an encrypted file container with a standard TrueCrypt volume (those are the default options).

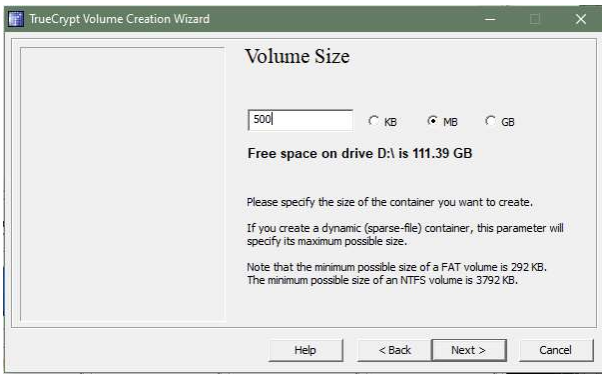


2. Click Select File and browse to a location where you want to create the new container. Make sure it is not in the Dropbox folder if Dropbox is running. You can name the container anyway you want, e.g. holiday2010.avi.



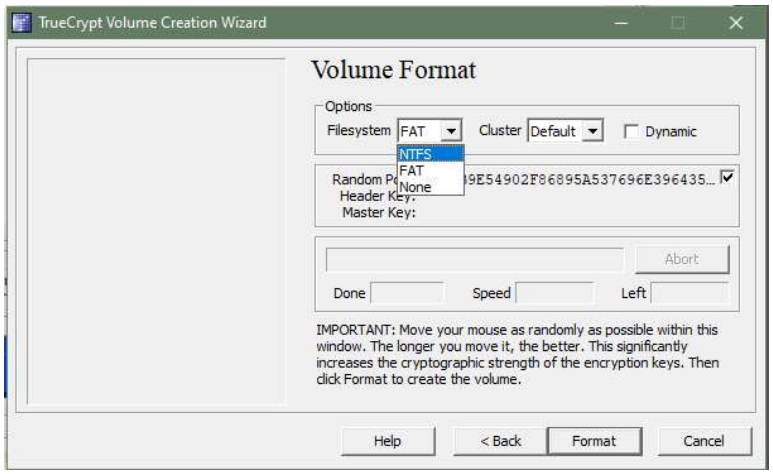
3. Click Next on the encryption options page unless you want to change the encryption algorithm or hash algorithm.

4. Select the volume size on the next screen. I suggest you keep it at a few hundred Megabytes tops.



5. You need to enter a secure password on the next screen. It is suggested to use as many characters as possible (24+) with upper and lower letters, numbers and special characters. The maximum length of a True Crypt password is 64 characters.

6. Now it is time to select the volume format on the next screen. If you only use Windows computers you may want to select NTFS as the file system. If you use others you may be better of with FAT. Juggle the mouse around a bit and click on format once you are done with that.



7. Congratulations, the new True Crypt volume has been created.