1.Write a python code to Network Packet Analysis with Scapy

```
pip install scapy
from scapy.all import
suspicious_ip =  192.168.0.6


def analyze_packet(packet):
    # Check if the packet has IP layer
    if IP in packet:
            if packet 192.168.0.8.src == suspicious_ip or packet[IP].dst == suspicious_ip:
           print(packet.summary())


print("Sniffing started.")
try:
    # Sniff packets with a filter for the suspicious IP address
    sniff(filter="host " + suspicious_ip, prn=analyze_packet, store=0)
except KeyboardInterrupt:
    print("Sn
```

2. Phishing Website Detection with Python

```
import requests
from bs4 import BeautifulSoup
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```
def extract_text_from_url(url):
    try:
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')
        text = ' '.join([p.text for p in soup.find_all('p')])
        return text
    except:
        print("Error fetching URL:", url)
        return None
phishing_url = "https://phishing1.com"
legitimate_url = "https://legitimate1.com"
phishing_text = extract_text_from_url(phishing_url)
legitimate_text = extract_text_from_url(legitimate_url)
if phishing_text is None or legitimate_text is None:
  exit()
phishing_data = [(phishing_text, 1) for _ in range(100)]
legitimate_data = [(legitimate_text, 0) for _ in range(100)]
data = phishing_data + legitimate_data
X, y = zip(*data)
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
test_urls = ["https://phishing.com", "https://legitimate.com"]
for url in test_urls:
    text = extract_text_from_url(url)
    if text is not None:
        features = vectorizer.transform([text])
        prediction = classifier.predict(features)[0]
        if prediction == 1:
            print(url, "is a phishing site.")
        else:
            print(url, "is a legitimate site.")
```