Phase 1: Research & Data Collection (Next Steps After Downloading Datasets)

Now that you've downloaded the datasets, the next step is **data preprocessing & feature extraction** to prepare the data for training your AI model. Here's what to do next:

---

1. Organize Your Dataset

First, check what kind of data you have:

- **PCAP files** (Network traffic captures)
- **CSV files** (Labeled network logs)
- **JSON/XML logs**

👉 **Create a structured folder for data management**:

```
mkdir -p project-root/data/raw project-root/data/processed
mv downloaded_datasets/* project-root/data/raw/
```

---

2. Explore the Data

Before processing, **understand the dataset structure**:

- If you have **CSV logs**, inspect them using Python:

```
import pandas as pd

df = pd.read_csv("project-root/data/raw/dataset.csv")
print(df.head())  # Show first few rows
print(df.info())  # Check data types and missing values
```

- If you have **PCAP files**, use Wireshark or convert them to CSV:

```
tshark -r dataset.pcap -T fields -E separator=, -e ip.src -e ip.dst -e
frame.time -e tcp.flags > dataset.csv
```

---

3. Clean & Preprocess the Data

Steps to follow:
✅ **Remove duplicate or missing values**
✅ **Convert timestamps into standard format**
✅ **Normalize numerical features** (e.g., packet size, request rate)
✅ **Label encoding** for categorical features (e.g., attack type)

Example: Cleaning a CSV dataset in Python

```
df.drop_duplicates(inplace=True)
df.fillna(0, inplace=True)  # Replace missing values with 0
df['timestamp'] = pd.to_datetime(df['timestamp'])  # Convert timestamps
```

---

4. Extract Key Features for AI Model Training

Now, extract meaningful features for attack detection, such as:

- **Traffic Flow Features** → Packet count, byte size, duration
- **Connection Features** → Source & destination IP, ports, protocol
- **Behavioral Features** → Request frequency, response time
- **Anomaly Indicators** → Sudden spikes, repetitive connection attempts

Example: Feature extraction using Scikit-learn

```
from sklearn.preprocessing import StandardScaler

features = ['packet_count', 'byte_size', 'duration']
scaler = StandardScaler()
df[features] = scaler.fit_transform(df[features])
```

---

5. Split Data into Training & Testing Sets

Once cleaned, split the dataset for model training:

```
from sklearn.model_selection import train_test_split

X = df.drop(columns=['attack_type'])  # Features
y = df['attack_type']  # Labels

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

---

6. Proceed to Phase 2: AI Model Training

Once preprocessing is done, we move to **Phase 2: Model Training**, where we:
✅ Choose an AI model (Random Forest, SVM, LSTM, etc.)
✅ Train the model on extracted features
✅ Test & evaluate accuracy

🚀 **Let me know when your data is preprocessed so we can move forward!**