

ANALYZING LOGS DATA WITH AI

MCS 2025 Cyber 242

Lecturer: Dominic Ligot

OBJECTIVES

- Understand the nature of log data
- Learn how to transform log data for analysis
- Apply various analysis techniques to find anomalies in log data

UNDERSTANDING LOG DATA

PACKET CAPTURE LOG (WIRESHARK)

The image displays a Wireshark packet capture log for interface *enp0s3. The packet list shows several DNS transactions. The selected packet (No. 429) is a DNS Standard query from 192.168.8.10 to 8.8.4.4 for www.google.com. The packet details pane shows the structure of the DNS query, including the Transaction ID (0x0eb0), Flags (0x0100), and the query for www.google.com (type A, class IN). The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
429	4.599509274	192.168.8.10	8.8.4.4	DNS	74	Standard query 0x0eb0 A www.google.com
430	4.599535582	192.168.8.10	8.8.4.4	DNS	74	Standard query 0x1ebf AAAA www.google.com
450	4.604775549	8.8.4.4	192.168.8.10	DNS	139	Standard query response 0x76ac A push.services.mozilla.com CN...
458	4.605231564	8.8.4.4	192.168.8.10	DNS	205	Standard query response 0x73a9 AAAA push.services.mozilla.com...
471	4.622154867	192.168.8.10	8.8.4.4	DNS	85	Standard query 0x6689 A getpocket.cdn.mozilla.net
472	4.622177628	192.168.8.10	8.8.4.4	DNS	85	Standard query 0x458a AAAA getpocket.cdn.mozilla.net
488	4.625366655	8.8.4.4	192.168.8.10	DNS	90	Standard query response 0x0eb0 A www.google.com A 172.217.14...

Frame 429: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s3, id 0

- Ethernet II, Src: PcsCompu_82:75:df (08:00:27:82:75:df), Dst: Ignition_01:f6:50 (00:78:cd:01:f6:50)
- Internet Protocol Version 4, Src: 192.168.8.10, Dst: 8.8.4.4
- User Datagram Protocol, Src Port: 58029, Dst Port: 53
- Domain Name System (query)
 - Transaction ID: 0x0eb0
 - Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
 - Queries
 - www.google.com: type A, class IN

[Response In: 488]

0000 00 78 cd 01 f6 50 08 00 27 82 75 df 08 00 45 00 ·x···P··'·u···E·
0010 00 3c 6c 3c 40 00 40 11 f9 b6 c0 a8 08 0a 08 08 ·<l<@·@······
0020 04 04 e2 ad 00 35 00 28 d4 f7 0e b0 01 00 00 01 ····5·(······

wireshark_enp0s3_20200427055131_kGXclt.pcapng Packets: 3286 · Displayed: 442 (13.5%) · Dropped: 0 (0.0%) Profile: Default

DATA AVAILABLE IN PACKET LOGS

- Timestamp
- Source IP
- Destination IP
- Protocol
- Source Port
- Destination Port
- Flags

LIMITATIONS OF LOG DATA

- Snapshot of network activity
- Limited context
- High volume and velocity
- Noise and redundancy
- Data quality issues
- Lack of standardization
- Retention limits
- Privacy and security concerns

THINGS TO BEAR IN MIND

- Data analysis does not replace domain knowledge – but complements it
- Data analysis is better at generating questions – but it remains the duty of the security analyst to put a case together
- Data analysis is about pattern recognition – look for similarities and anomalies as clues to formulating a hypothesis

SETTING UP GOOGLE COLLAB

GITHUB REPOSITORY

- Access/clone/fork from here:

PCAP CONVERTER NOTEBOOK

The screenshot displays a Jupyter Notebook interface for a file named `PCAP_Converter.ipynb`. The top navigation bar includes a menu with `File`, `Edit`, `View`, `Insert`, `Runtime`, `Tools`, and `Help`, along with a status indicator `All changes saved`. On the right of the top bar are links for `Comment`, `Share`, and a settings icon.

The left sidebar features a `Files` panel showing a directory structure:

- `..`
- `.config`
- `.ipynb_checkpoints`
- `Exercise 1`
- `Exercise 3`
- `sample_data`

The main content area is titled `PCAP Log Data Analysis` and contains the following text:

Compiling some helper functions to help with PCAP Log analysis:

- Convert PCAP to CSV
- Network activity analysis
- Graph theory analysis
- Clustering: One hot encoding, Dimensional Reduction, Clustering

Below this text is a code cell with the following content:

```
[55] !pip install scapy
```

The output of the code cell shows:

```
Requirement already satisfied: scapy in /usr/local/lib/python3.10/dist-packages (2.6.0)
```

At the bottom of the code cell, there is a section titled `# Convert PCAP to CSV` with the following Python code:

```
from scapy.all import rdpcap
import csv
import socket
from datetime import datetime
```

The bottom status bar indicates `Disk` usage and `75.39 GB available`.

PCAP CONVERTER NOTEBOOK

- Scripts are provided for the following functions:
 - Convert PCAP to CSV
 - Network Activity Analysis
 - Graph Theory Analysis
 - Clustering Analysis:
 - One-hot-encoding
 - Dimensional Reduction
 - Clustering















PCAP LAB EXERCISES

PCAP LAB EXERCISES

- 3 exercises are provided
- Exercise 1 will have forensic analysis, alerts, and the PCAP file
- Exercises 2 and 3 will only have the PCAP file. Forensic analysis and alerts will be provided after.

DATA ANALYSIS OF PCAP

TIME SERIES – EYEBALLING

Count of Packet_Number	Column Labels 			
Row Labels 	172.17.0.17	172.17.0.99	79.124.78.197	Grand Total
 :00	4	43	4	51
 :01	2	17	6	25
 :02	10	34	4	48
 :03	4	6	1	11
 :04	2	31	5	38
 :05	2	23	7	32
 :06	2	10		12
 :07	1	29	9	39
 :08	1	12	5	18
 :09	1	5	2	8
 :10	3	9	4	16
 :11	1	7	6	14

IP TO PORT - EYEBALLING

Count of Packet_Number	Column Labels ▼												
Row Labels ▼	53	67	68	80	88	123	135	137	138	139	389	443	
23.45.119.143													14
23.45.119.144													199
23.45.119.147													13
40.119.249.228													22
40.126.28.12													20
40.126.28.22													11
46.254.34.201													504
52.109.0.142													13
52.109.0.91													16
52.113.194.132													73
79.124.78.197					261								
(blank)													
Grand Total	87	2	2	290	45	8	30	18	18	55	177	1536	

IP TO IP - EYEBALLING

Count of Packet_Number	Column Labels ▼			
Row Labels ▼	172.17.0.99	255.255.255.255	(blank)	Grand Total
23.45.119.147	13			13
40.119.249.228	22			22
40.126.28.12	20			20
40.126.28.22	11			11
46.254.34.201	504			504
52.109.0.142	13			13
52.109.0.91	16			16
52.113.194.132	73			73
79.124.78.197	261			261
(blank)			294	294
Grand Total	1826	2	294	2122

NETWORK ACTIVITY STATISTICS

```
=== Network Traffic Analysis Report ===
```

```
Basic Statistics:
```

```
total_packets: 5091
```

```
unique_ips: 42
```

```
unique_connections: 77
```

```
avg_packet_size: 423.1392537002293
```

```
duration_seconds: 3576.159984
```

```
Top Talkers:
```

```
|
```

```
Top Source IPs:
```

```
172.17.0.99: 2358 packets
```

```
172.17.0.17: 611 packets
```

```
46.254.34.201: 504 packets
```

```
79.124.78.197: 261 packets
```

```
23.45.119.144: 199 packets
```

```
23.221.24.69: 147 packets
```

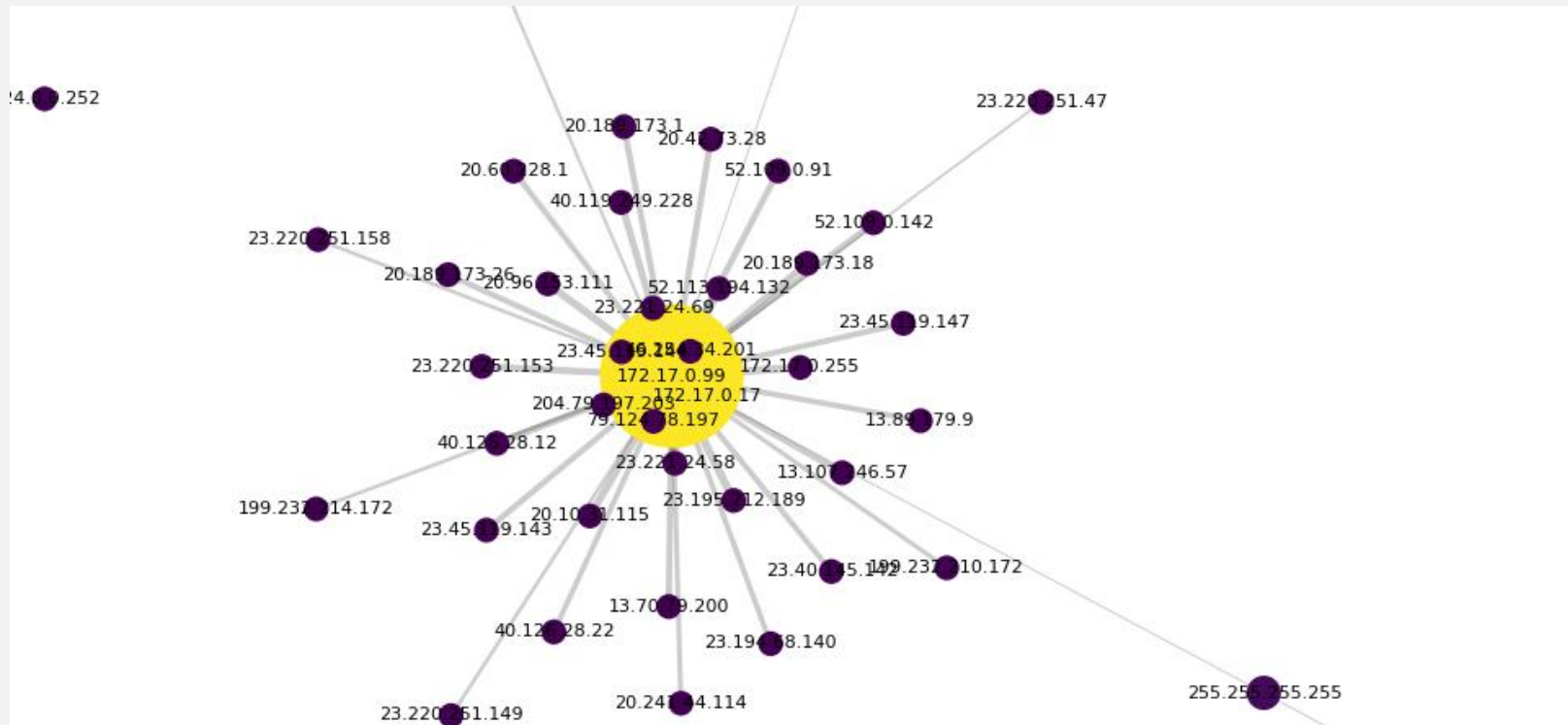
```
204.79.197.203: 144 packets
```

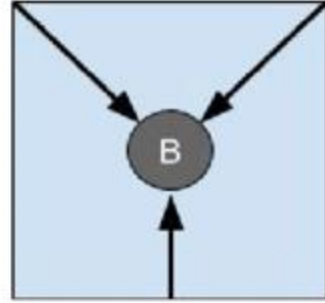
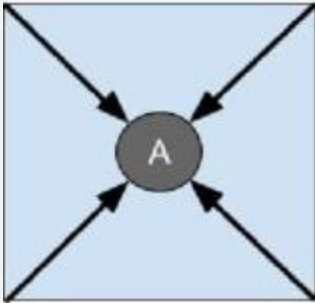
```
23.221.24.58: 91 packets
```

```
52.113.194.132: 73 packets
```

```
23.195.212.189: 37 packets
```

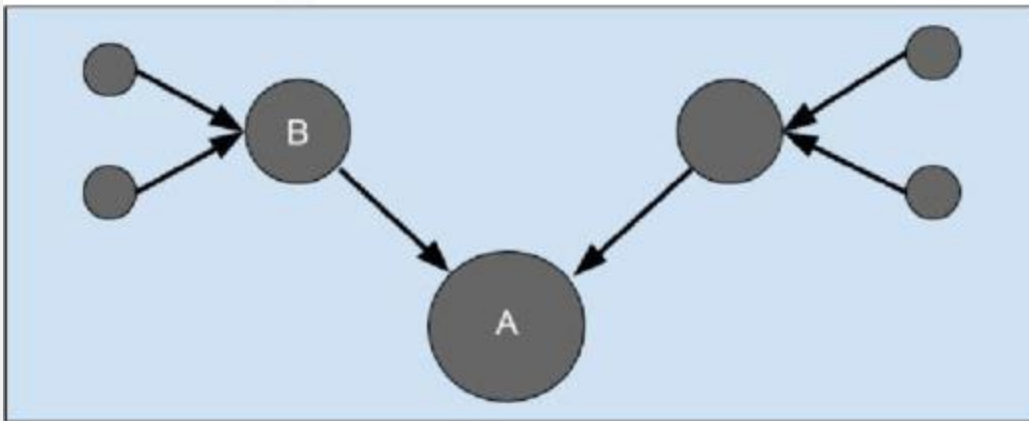
NETWORK VISUALIZATION





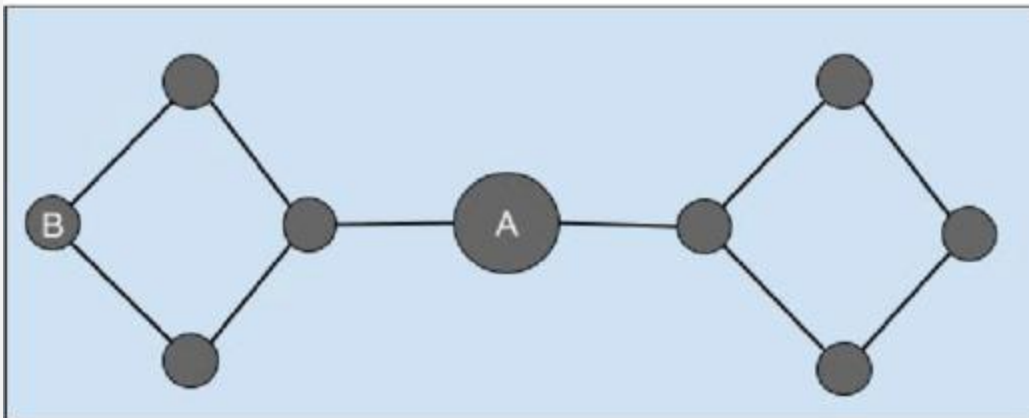
In-degree Centrality

User A has a higher in-degree centrality than user B because user A has more followers than user B.



Eigenvector Centrality

While users A and B both have the same in-degree centrality (two followers), user A has a higher Eigenvector centrality because the weight of the two followers is higher.



Betweenness Centrality

User A has a higher betweenness centrality than user B. A message sent from user A will reach many more users in a shorter path compared to user B.

NETWORK GRAPH ANALYSIS

=== Network Graph Analysis Report ===

Basic Graph Metrics:

nodes: 42
edges: 41
density: 0.047619047619047616
avg_clustering: 0.0
avg_shortest_path: 2.085946573751452
diameter: 4
avg_degree: 1.9523809523809523

Protocol Distribution:

UDP: 575 packets (11.99%)
TCP: 4222 packets (88.01%)

Most Important Nodes:

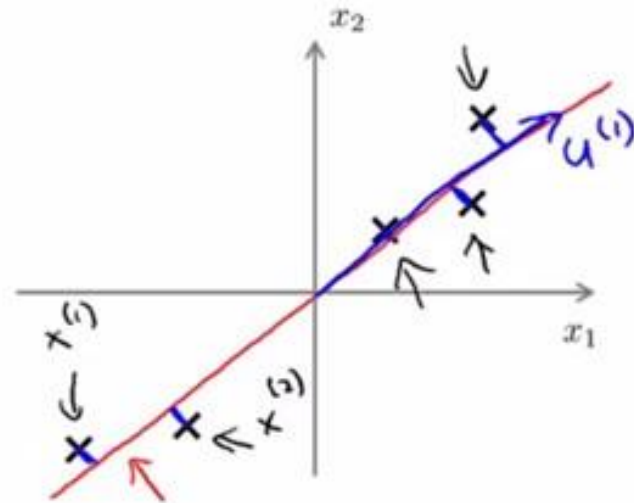
	degree centrality	betweenness centrality	eigenvector centrality	total_packets	importance_score
172.17.0.99	0.95122	0.996341	0.706847	4793.0	1198.913602
172.17.0.17	0.04878	0.095122	0.116203	1310.0	327.565026
46.254.34.201	0.02439	0.000000	0.113148	782.0	195.534384
79.124.78.197	0.02439	0.000000	0.113148	591.0	147.784384
23.45.119.144	0.02439	0.000000	0.113148	376.0	94.034384

ONE HOT ENCODING

Source_IP_0.0.0.0	Source_IP_13.107.246.57	Source_IP_13.70.79.200	Source_IP_13.89.179.9	Source_IP_172.170.17	Source_IP_172.170.99	Source_IP_184.29.137.96	Source_IP_199.232.10.172	Source_IP_199.232.14.172	Source_IP_20.10.31.115	Source_IP_20.10.31.115
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0

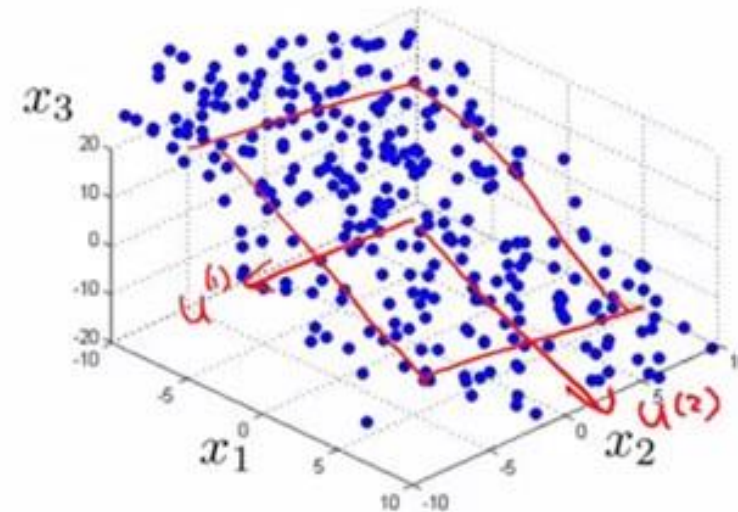
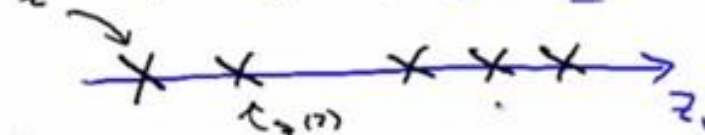
DIMENSIONAL REDUCTION

Principal Component Analysis (PCA) algorithm



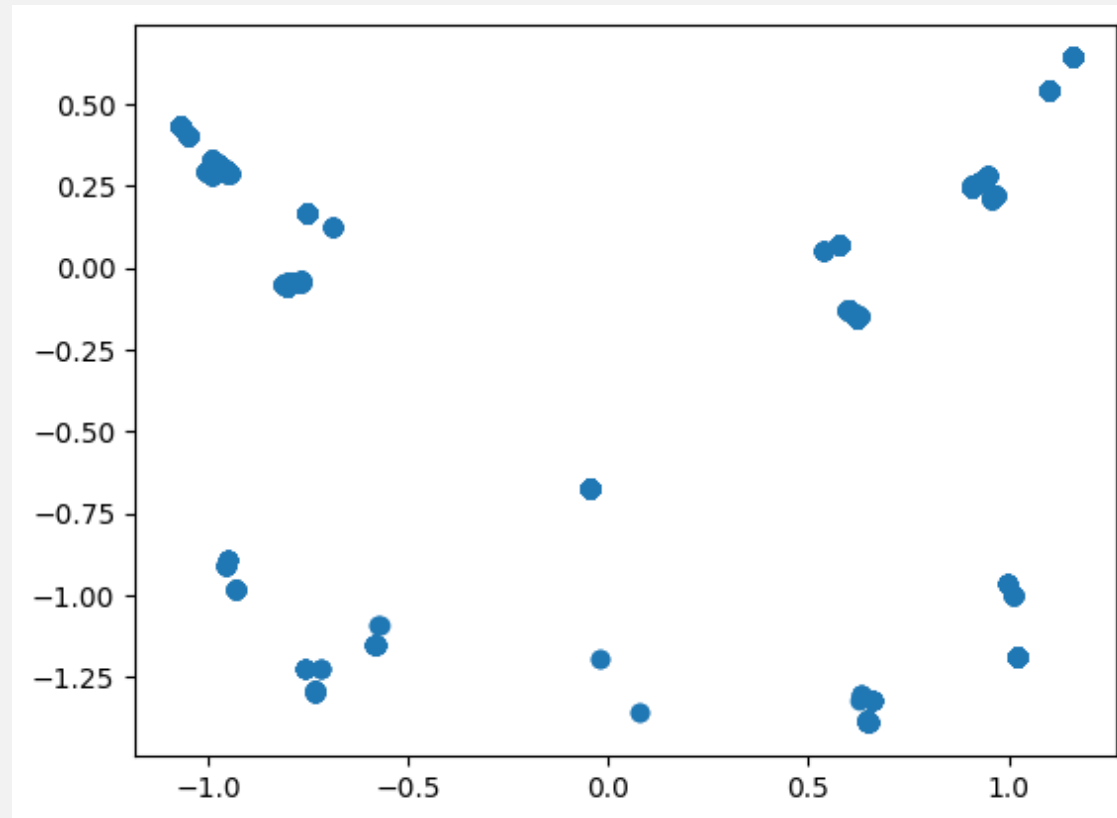
Reduce data from 2D to 1D

$$x^{(i)} \in \mathbb{R}^2 \rightarrow z^{(i)} \in \mathbb{R}$$



Reduce data from 3D to 2D

DIMENSIONAL REDUCTION

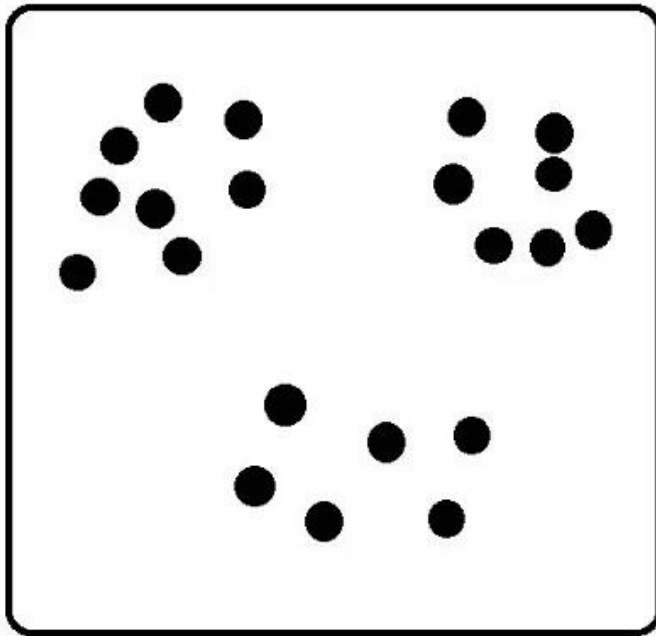


DIMENSIONAL REDUCTION

Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	Destinatic	x_pos	y_pos
0	0	0	0	0	0	0	0	0	0	-0.02114	-1.19303
0	0	0	0	0	0	0	0	0	0	0.078709	-1.35421
0	0	0	0	0	0	0	0	0	0	-0.02114	-1.19303
0	0	0	0	0	0	0	0	0	0	0.078709	-1.35421
0	0	0	0	0	0	0	0	0	0	-0.04392	-0.6761
0	0	0	0	0	0	0	0	0	0	-0.04392	-0.6761
0	0	0	0	0	0	0	0	0	0	-0.73347	-1.29116
0	0	0	0	0	0	0	0	0	0	-0.57256	-1.09145
0	0	0	0	0	0	0	0	0	0	0.647581	-1.38131
0	0	0	0	0	0	0	0	0	0	-0.57256	-1.09145
0	0	0	0	0	0	0	0	0	0	-0.57211	-1.08896
0	0	0	0	0	0	0	0	0	0	-0.73366	-1.29214
0	0	0	0	0	0	0	0	0	0	-0.73366	-1.29214
0	0	0	0	0	0	0	0	0	0	0.647753	-1.38236
0	0	0	0	0	0	0	0	0	0	0.647753	-1.38236

CLUSTERING

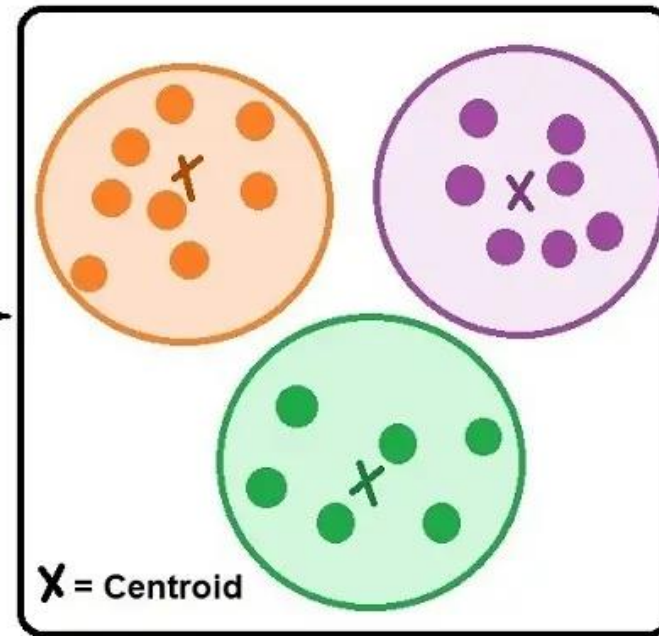
Unlabeled Data



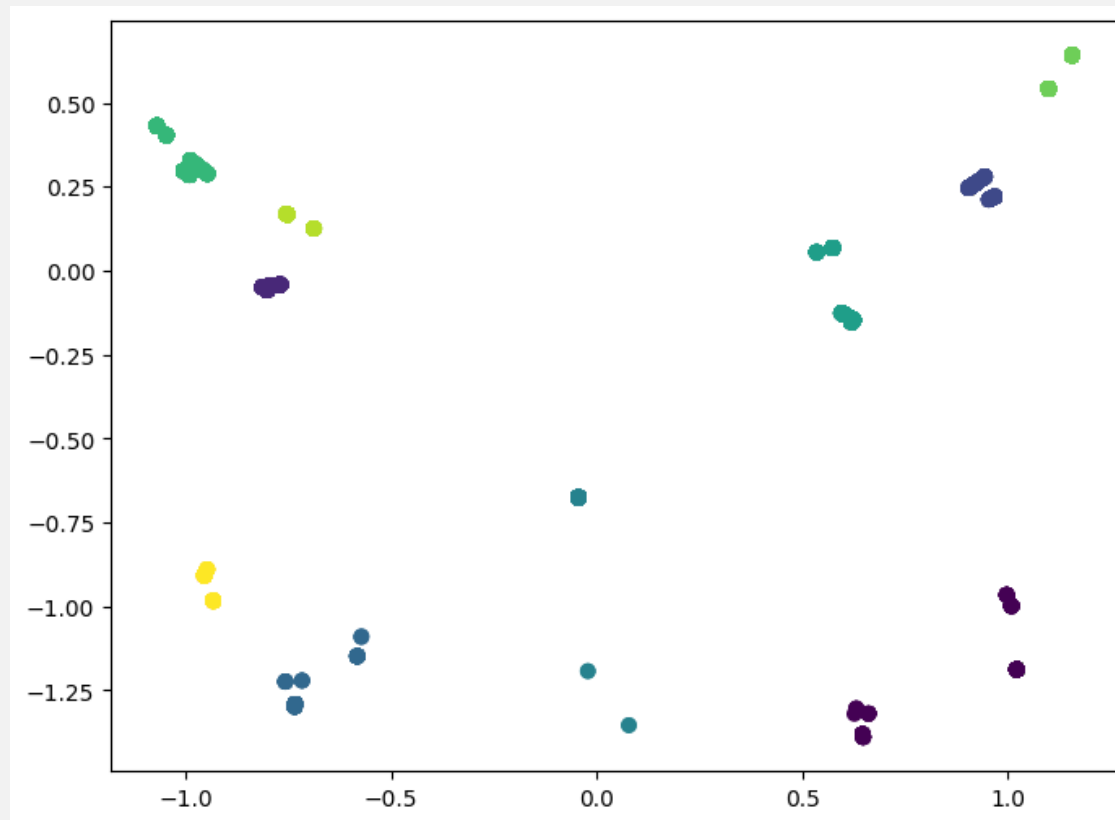
K-Means



Labeled Clusters



CLUSTERING



CLUSTERING

Count of Packet_Number	Column Labels <input type="button" value="v"/>										
Row Labels <input type="button" value="v"/>	0	1	2	3	4	5	6	7	8	9	Grand Total
40.119.249.228			22								22
40.126.28.12			20								20
40.126.28.22			11								11
46.254.34.201								504			504
52.109.0.142			13								13
52.109.0.91			16								16
52.113.194.132			73								73
79.124.78.197						261					261
(blank)					294						294
Grand Total	320	599	814	138	298	797	1144	504	364	113	5091

RECAP AND Q&A

OBJECTIVES

- Understand the nature of log data
- Learn how to transform log data for analysis
- Apply various analysis techniques to find anomalies in log data
 - Time Series
 - Matching
 - Network Activity
 - Graph Network Analysis
 - Clustering

ANALYZING LOGS DATA WITH AI

MCS 2025 Cyber 242

Lecturer: Dominic Ligot