Introducing UWF-ZeekData: Network Datasets Based on the MITRE ATT&CK Framework

Dustin Mink, Sikha Bagui & Subhash Bagui

University of West Florida, USA

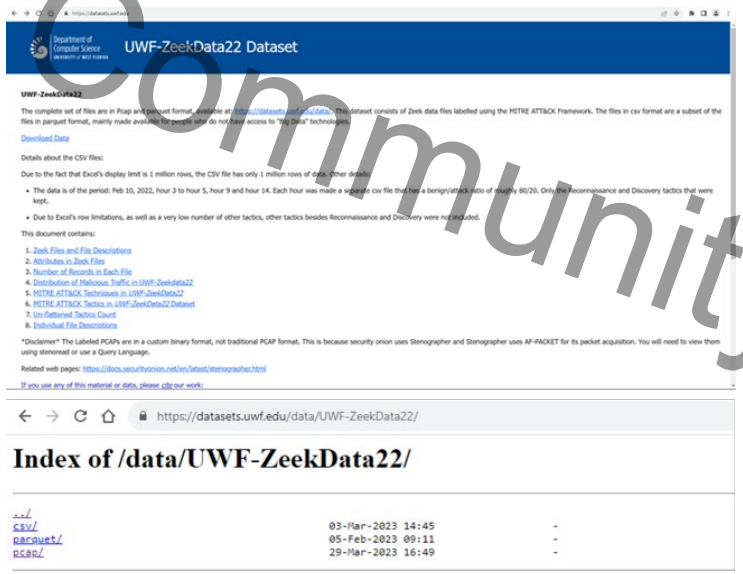
Agenda

- Introduction
- Cyber Analytics Research Group
- Data
- Correlation
- Machine Learning
- Questions

Introduction

- UWF-ZeekData22 is the first network instruction detection dataset of Zeek logs labelled with the MITRE ATT&CK framework
- University of West Florida's (UWF) Cyber Analytics Research Group (CAR) is an interdisciplinary research group

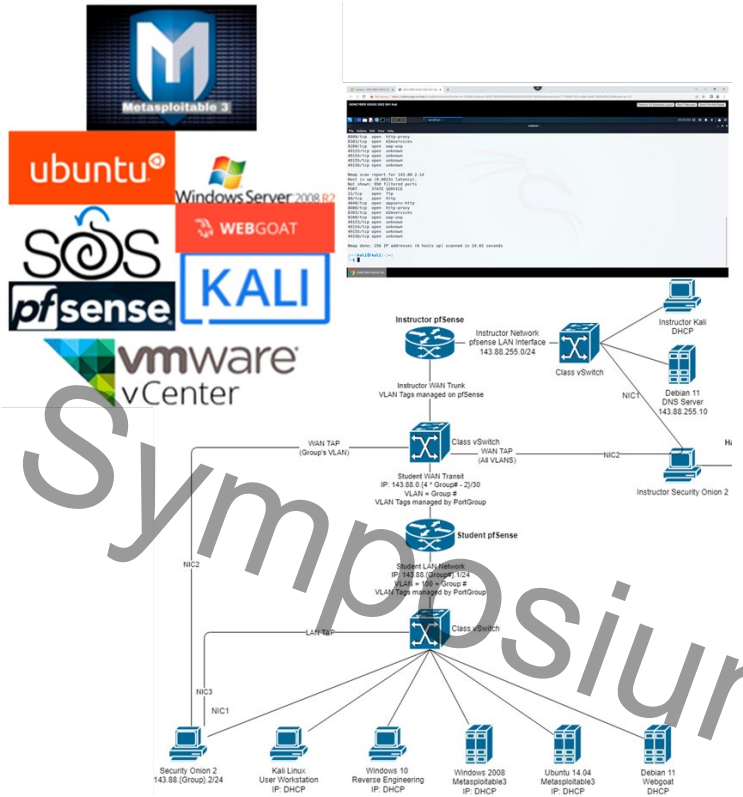
Datasets



- Publically available web site
- Zeek and MITRE ATT&CK in CSV and Parquet formats
- Raw network traffic in PCAP

UWF Cyber Range

- (x5) Dell PowerEdge R750 (128CPU 1TB RAM 85TB SSD, Tesla T4 GPU)
- (x2) Dell PowerEdge R740 (48CPU 768 GB RAM 13TB SSD)
- VMware vCenter
- VMware PowerCLI
- Kali, Security Onion, Pfsense, Metasplotiable 3 (Windows/ Ubuntu), WebGoat



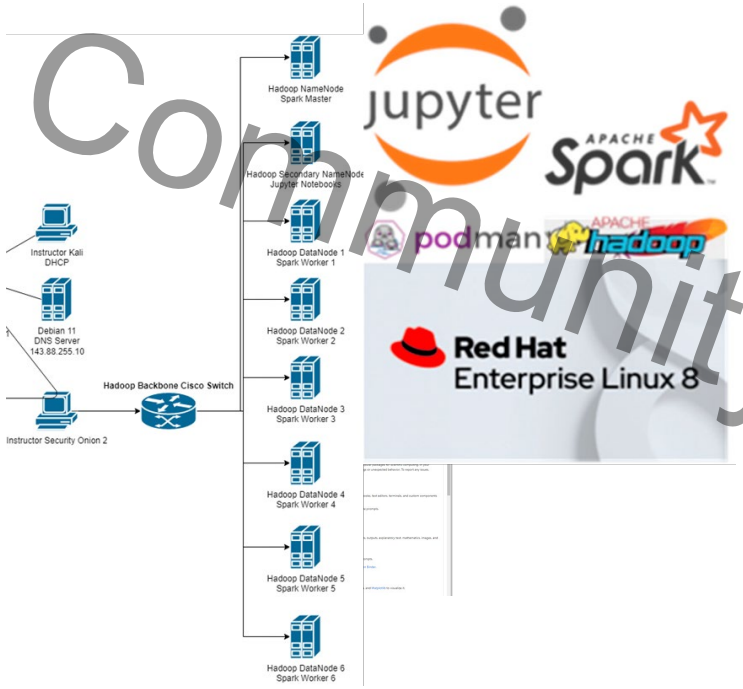
Ethical Hacking and Penetration Testing

- CAE-CD Cybersecurity Program
- Each student has their own Kali VM
- Victims Windows Metasploitable 3, Ubuntu Metasploitable 3, and Ubuntu WebGoat

Cyber War Gaming

- CAE-CD Cybersecurity Program
- Pairs of Student team up for Capture the Flag
- Kali and Security Onion VMs

UWF Big Data Platform



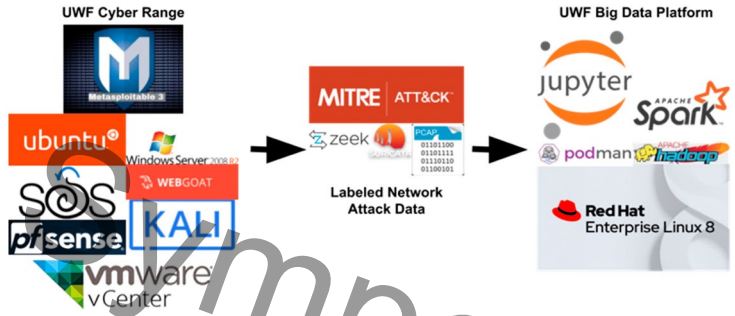
- (x5) Dell PowerEdge R750 (128CPU 1TB RAM 85TB SSD, Tesla T4 GPU)
- Hadoop, Spark, Jupyter Notebooks

Big Data Class

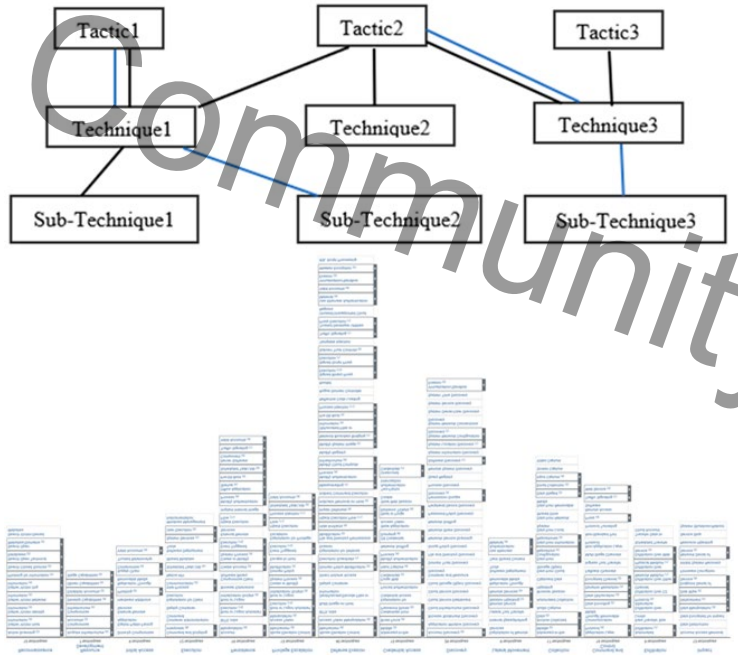
- Undergraduate and graduate degrees in Computer Science;
- Graduate degree in Data Science
- Concepts of Hadoop and MapReduce are covered
- Big Data programming using Spark is introduced

Zeek, PCAP, and Mission Logs

- Instructure Security Onion 2 Collects Zeek and PCAPs
- CronTab runs Bach Script to transfer Zeek and PCAPs to HDFS
- Student enter metadata into Google Form
- End of the semester transfer mission logs to HDFS



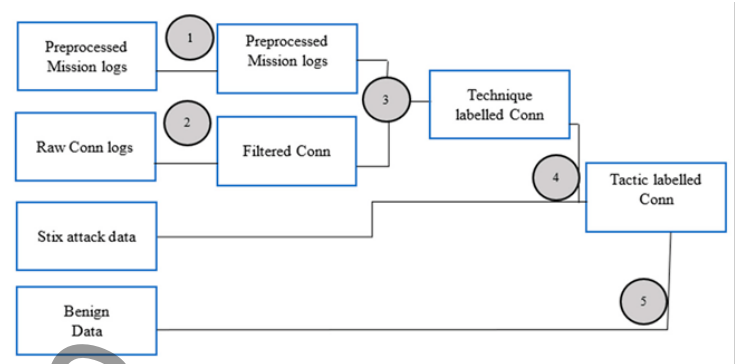
MITRE ATT&CK

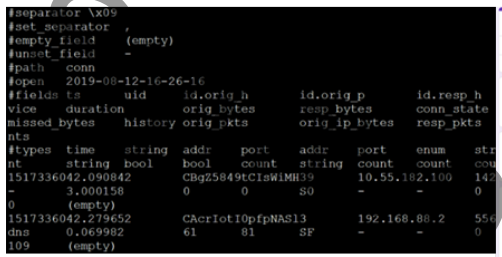


- MITRE ATT&CK cybersecurity industry standard
- UWF-ZeekData22, has 14 tactics,191 techniques, and 358 sub-techniques

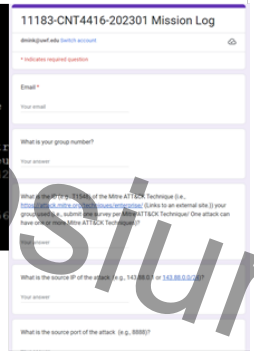
Correlate Zeek and Mission Logs

- Zeek Log
- Mission Log
- Start datetime, stop datetime, src ip, dest ip, src port, and dest port





```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2019-08-12-16-26-16
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h
vice    duration        orig_bytes      resp_bytes      conn_state
missed_bytes    history orig_pkts       orig_ip_bytes   resp_pkts
nts
#types  time    string  addr    port    addr    port    enum    str
nt      string  bool    bool    count   string  count   count   cou
1517336042.090842       CBgZ5849tCIsWiMH39      10.55.182.100   142
-       3.000158        0       0       S0      -       -       0
0       (empty)
1517336042.279652       CAcrIotI0pfpNAS13       192.168.88.2    554
dns     0.069982        61      81      SF      -       -       0
109     (empty)
```

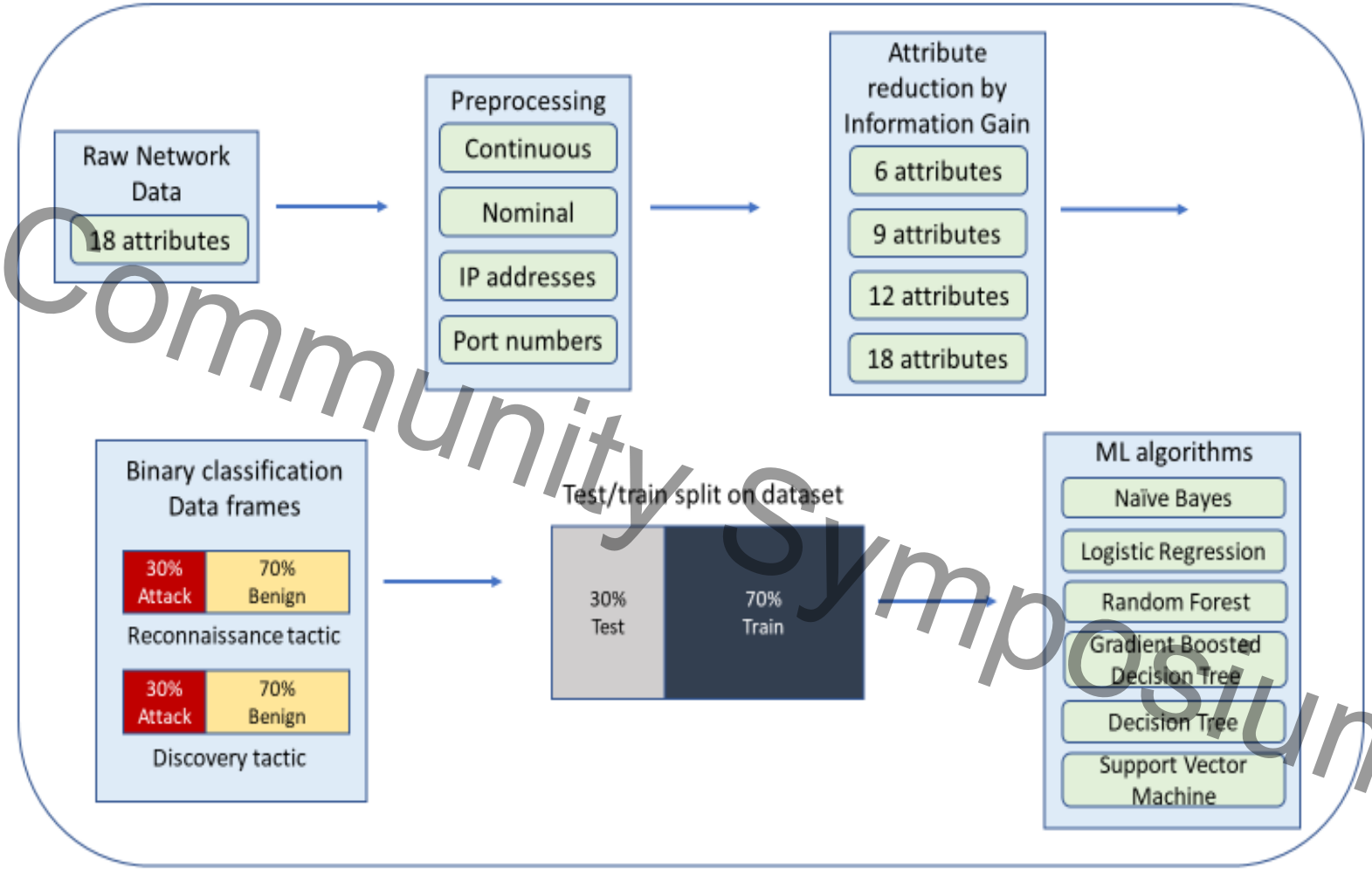


UWF-ZeekData22 Data Schema


```
>>> df_conn.printSchema()
root
 |-- resp_pkts: integer (nullable = true)
 |-- mitre_attack: string (nullable = true)
 |-- service: string (nullable = true)
 |-- orig_ip_bytes: integer (nullable = true)
 |-- local_resp: boolean (nullable = true)
 |-- missed_bytes: integer (nullable = true)
 |-- proto: string (nullable = true)
 |-- duration: double (nullable = true)
 |-- conn_state: string (nullable = true)
 |-- dest_ip_zeek: string (nullable = true)
 |-- orig_pkts: integer (nullable = true)
 |-- community_id: string (nullable = true)
 |-- resp_ip_bytes: integer (nullable = true)
 |-- dest_port_zeek: integer (nullable = true)
 |-- orig_bytes: integer (nullable = true)
 |-- local_orig: boolean (nullable = true)
 |-- datetime: timestamp (nullable = true)
 |-- history: string (nullable = true)
 |-- resp_bytes: integer (nullable = true)
 |-- uid: string (nullable = true)
 |-- src_port_zeek: integer (nullable = true)
 |-- ts: double (nullable = true)
 |-- src_ip_zeek: string (nullable = true)
```

- Zeek Connection Log
- MITRE ATT&CK Tactic

Machine Learning



Determining Spark's Optimum Parameters

Test ID	Executor Count	Cores Per Executor	Memory Per Executor	Total Executor Cores	Total Executor Memory (GB)	Total Time (seconds)	Shuffle partitions	Driver Cores	Driver memory (GB)
1	5	2	5	10	25	355.24	200	2	10
2	5	2	10	10	50	346.30	200	2	10
3	5	2	20	10	100	337.45	200	2	10
4	5	4	5	20	25	283.60	200	2	10
5	5	4	10	20	50	276.21	200	2	10
6	5	4	20	20	100	277.28	200	2	10
7	10	2	5	20	50	283.18	200	2	10
8	10	2	10	20	100	276.20	200	2	10
9	10	4	5	40	50	210.99	200	2	10
10	10	4	10	40	100	199.78	200	2	10
11	20	2	5	40	100	214.43	200	2	10
12	20	4	5	80	100	186.02	200	2	10
13	10	8	10	80	100	175.59	200	2	10
14	12	8	10	96	120	172.91	200	2	10
16	12	8	10	96	120	171.49	72	2	10
17	12	8	10	96	120	164.35	12	2	10
18	24	4	5	96	120	183.51	24	2	10
19	6	16	20	96	120	162.02	6	2	10
20	12	8	10	96	120	170.18	24	2	10
21	3	32	40	96	120	168.58	3	2	10
22	1	16	20	16	20	243.16	1	2	10
23	2	16	20	32	40	210.14	2	2	10
24	6	32	40	192	240	155.39	6	2	10
25	3	16	20	48	60	183.75	3	2	10
26	10	8	10	80	100	178.59	200	2	10
27	6	32	40	192	240	156.8	6	2	10
28	6	32	40	192	240	161.84	12	2	10
29	6	32	40	192	240	159.12	24	2	10
30	6	32	40	192	240	159.31	48	2	10
31	6	32	40	192	240	159.91	96	2	10
32	6	32	40	192	240	161.13	192	2	10
33	6	32	40	192	240	161.6	384	2	10
34	6	32	40	192	240	159.2	6	4	10
35	6	32	40	192	240	157.79	6	4	20
36	6	32	40	192	240	158.2	6	4	30

Reconnaissance: Accuracy – by Algorithms by Number of Features

ML Algo	Attr.	Accuracy	Precision	Recall	F-measure	AUROC	FPR	Training	Testing
DT	6	99.30%	99.09%	98.58%	98.84%	99.10%	0.39%	27.933	0.087
DT	9	99.31%	99.10%	98.60%	98.85%	99.11%	0.39%	28.878	0.088
DT	12	99.35%	99.20%	98.65%	98.92%	99.15%	0.34%	29.75	0.086
DT	18	99.40%	99.69%	98.30%	98.99%	99.08%	0.13%	28.365	0.071
GBT	6	99.26%	99.39%	99.56%	99.48%	99.07%	1.42%	80.639	0.077
GBT	9	99.29%	99.39%	99.60%	99.50%	99.09%	1.42%	80.178	0.076
GBT	12	99.30%	99.38%	99.62%	99.50%	99.08%	1.46%	79.599	0.075
GBT	18	99.37%	99.23%	99.88%	99.55%	99.03%	1.81%	59.147	0.087
LR	6	96.52%	94.02%	94.38%	94.20%	95.91%	2.57%	22.1	0.057

Reconnaissance: Accuracy – by Algorithms by Number of Features Cont.

ML Algo	Attr.	Accuracy	Precision	Recall	F-measure	AUROC	FPR	Training	Testing
LR	6	96.52%	94.02%	94.38%	94.20%	95.91%	2.57%	22.1	0.057
LR	9	96.52%	94.02%	94.38%	94.20%	95.91%	2.57%	22.265	0.051
LR	12	96.52%	94.02%	94.38%	94.20%	95.91%	2.57%	22.372	0.051
LR	18	96.52%	94.02%	94.38%	94.20%	95.91%	2.57%	23.375	0.052
NB	6	95.84%	92.11%	94.19%	93.14%	95.37%	3.46%	15.634	0.053
NB	9	95.85%	92.11%	94.22%	93.15%	95.38%	3.46%	16.078	0.091
NB	12	95.85%	92.11%	94.21%	93.15%	95.38%	3.46%	15.7	0.062
NB	18	95.86%	92.12%	94.27%	93.18%	95.41%	3.46%	15.234	0.056

Reconnaissance: Accuracy – by Algorithms by Number of Features Cont.

ML Algo	Attr.	Accuracy	Precision	Recall	F-measure	AUROC	FPR	Training	Testing
RF	6	99.19%	98.95%	99.90%	99.42%	98.72%	2.47%	56.257	0.048
RF	9	98.11%	97.39%	99.98%	98.67%	96.86%	6.26%	56.276	0.075
RF	12	99.19%	98.92%	99.94%	99.43%	98.70%	2.55%	56.473	0.052
RF	18	99.22%	98.94%	99.96%	99.45%	98.73%	2.51%	47.286	0.054
SVM	6	70.01%	0.00%	0.00%	0.00%	50.00%	0.00%	39.053	0.031
SVM	9	96.87%	95.23%	94.28%	94.75%	96.13%	2.02%	68.317	0.036
SVM	12	97.36%	97.08%	94.02%	95.53%	96.41%	1.21%	64.397	0.036
SVM	18	97.93%	99.04%	94.00%	96.45%	96.80%	0.39%	66.216	0.036

Conclusions: Optimizing Classifier Performance on Spark

- More total cores for spark application makes ML algorithms run faster, but there are diminishing returns after a certain point
- Classifiers run fastest when the number of shuffle partitions is the same as the total number of executors

There was no significant correlation between runtimes and the total amount of memory allocated (though allocating too little memory can cause executors to crash)

Conclusions: Machine Learning Results

- Tree-based methods (DT, GBT, RF) performed better on most metrics than the other three algorithms in classifying this dataset, for both the Renaissance and Discovery tactics
  - These three algorithms all showed 99%+ accuracy for both attack tactics, with similarly higher scores in precision, recall, f-measure, and AUROC.
  - GBT and RF performed a little better than DT in terms of recall for both the tactics but in terms of the FPR
  - DT had the lowest FPRs for both Reconnaissance and Discovery

Conclusions: Machine Learning Results Cont'

- Training times -- RF performed the best for Reconnaissance, followed by DT
- For Discovery, DT performed the best
- Best number of features -- the top 6 features from information gain:
  - history
  - protocol
  - service
  - orig_bytes
  - dest_ip
  - orig_pkts

Conclusion

- UWF-ZeekData22 is the first network instruction detection dataset of Zeek logs labelled with the MITRE ATT&CK framework

Acknowledgements

- The Research was supported by 2021 NCAE-C-002: Cyber Research Innovation Grant Program, Grant Number: H98230-21-1-0170

2024 CAE Community Symposium

Questions?

Data available at: https://datasets.uwf.edu