CIS 660 - FINAL PROJECT

AWID ANOMALY DETECTION

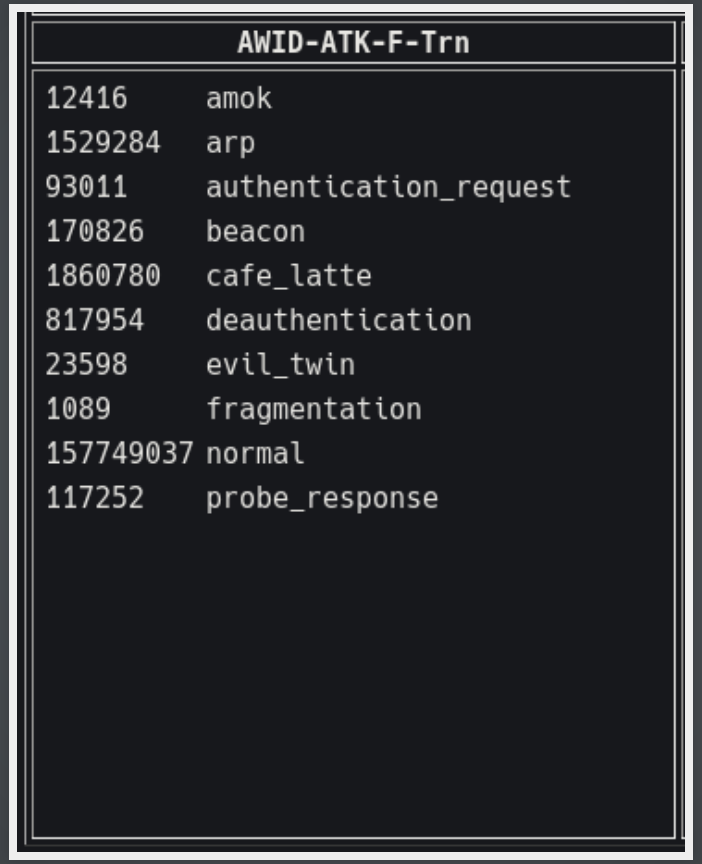
TEAM

- Paul Webster
- Gabriel Madison
- Brandon Marlowe
- Mirza Baig

AWID (AEGEAN WI-FI INTRUSION) DATASET

DATASET PRODUCED FROM REAL WIRELESS NETWORK LOGGING

FULL DATASET



AWID-ATK-F-Trn

12416	amok
1529284	arp
93011	authentication_request
170826	beacon
1860780	cafe_latte
817954	deauthentication
23598	evil_twin
1089	fragmentation
157749037	normal
117252	probe_response

- ~ 160,000,000 Rows x 155 Columns

REDUCED TRAINING SET USED

AWID-ATK-R-Trn	
31180	amok
64609	arp
3500	authentication_request
1799	beacon
45889	cafe_latte
10447	deauthentication
2633	evil_twin
770	fragmentation
1633190	normal
1558	probe_response

- ~ 1.8 Million Rows x 155 Columns
- Produced from 1 hour of logging

MAJORITY OF DATA IS OF "NORMAL" CLASS IN EITHER DATASET

PROJECT GOAL

BUILD A CLASSIFIER CAPABLE OF PROPERLY CLASSIFYING TUPLES WITH FOUR SPECIFIC ATTACK TYPES:

Amok
Deauthentication
Authentication Request
ARP

3 MAJOR TASKS

- Preprocessing/Cleaning
- Feature Selection
- Classification

ABOUT THE ATTACKS

DEAUTHENTICATION

A DENIAL OF SERVICE ATTACK THAT USES UNPROTECTED DEAUTHENTICATION PACKETS TO SPOOF AN ENTITY. THE ATTACKER MONITORS TRAFFIC ON A NETWORK TO DISCOVER MAC ADDRESSES ASSOCIATED WITH SPECIFIC CLIENTS. A DEAUTHENTICATION MESSAGE IS THEN SENT TO THE ACCESS POINT ON BEHALF OF A PARTICULAR MAC ADDRESS, WHICH FORCES THAT CLIENT OFF THE NETWORK. THE ATTACKER THEN CONNECTS TO THE ACCESS POINT AS THE CLIENT THAT WAS PREVIOUSLY DISCONNECTED.

AUTHENTICATION REQUEST

A TYPE OF FLOODING ATTACK -> "IN THIS CASE THE AGGRESSOR ATTEMPTS TO EXHAUST THE AP'S RESOURCES BY CAUSING OVERFLOW TO ITS CLIENT ASSOCIATION TABLE. IT IS BASED ON THE FACT THAT THE MAXIMUM NUMBER OF CLIENTS WHICH CAN BE MAINTAINED IN THE CLIENT AP'S ASSOCIATION TABLE IS LIMITED AND DEPENDS EITHER ON A HARD-CODED VALUE ON THE AP OR ON ITS PHYSICAL MEMORY CONSTRAINTS. AN ENTRY ON THE AP'S CLIENT ASSOCIATION TABLE IS INSERTED UPON THE RECEIPT OF AN AUTHENTICATION REQUEST MESSAGE EVEN IF THE CLIENT DOES NOT COMPLETE ITS AUTHENTICATION (I.E., IS STILL IN THE UNAUTHENTICATED/UNASSOCIATED STATE)." - INTRUSION DETECTION IN 802.11 NETWORKS: EMPIRICAL EVALUATION OF THREATS AND A PUBLIC DATASET

AMOK

ANOTHER FLOODING ATTACK, SIMILAR TO AUTHENTICATION REQUEST

ARP (ADDRESS RESOLUTION PROTOCOL)

"IN COMPUTER NETWORKING, ARP SPOOFING, ARP CACHE POISONING, OR ARP POISON ROUTING, IS A TECHNIQUE BY WHICH AN ATTACKER SENDS (SPOOFED) ADDRESS RESOLUTION PROTOCOL (ARP) MESSAGES ONTO A LOCAL AREA NETWORK. GENERALLY, THE AIM IS TO ASSOCIATE THE ATTACKER'S MAC ADDRESS WITH THE IP ADDRESS OF ANOTHER HOST, SUCH AS THE DEFAULT GATEWAY, CAUSING ANY TRAFFIC MEANT FOR THAT IP ADDRESS TO BE SENT TO THE ATTACKER INSTEAD." - WIKIPEDIA

PREPROCESSING/CLEANING

STARTING POINT

```
0  ?  0.000000000  1393661302.645757000  0.000000000  0.000000000  0.000000000     261   261   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.670028000  0.024271000  0.024271000  0.024271000     185   185   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.671659000  0.001631000  0.001631000  0.025902000     185   185   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.726984000  0.055325000  0.055325000  0.081227000     159   159   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.727399000  0.000415000  0.000415000  0.081642000     54    54    0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.727404000  0.000005000  0.000005000  0.081647000     40    40    0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.744096000  0.016692000  0.016692000  0.098339000     261   261   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.744238000  0.000142000  0.000142000  0.098481000     40    40    0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
0  ?  0.000000000  1393661302.772305000  0.028067000  0.028067000  0.126548000     185   185   0  0  0  0  26  1  1  1  1  0  1  0  0  0  0  0  1  0  0  1  0  0  0  0  0x00000000  0  0  0  21
```

WIRESHARK COLUMN NAMES

```
frame.interface_id
frame.dlt
frame.offset_shift
frame.time_epoch
frame.time_delta
frame.time_delta_displayed
frame.time_relative
frame.len
frame.cap_len
frame.marked
frame.ignored
radiotap.version
radiotap.pad
radiotap.length
radiotap.present.tsft
radiotap.present.flags
radiotap.present.rate
radiotap.present.channel
radiotap.present.fhss
radiotap.present.dbm_antsignal
radiotap.present.dbm_antnoise
radiotap.present.lock_quality
radiotap.present.tx_attenuation
```

ADDING WIRESHARK COLUMN NAMES

```
[FILE: col_names.txt]

frame.interface_id
frame.dlt
frame.offset_shift

...

wlan.qos.buf_state_indicated
data.len
class
```

```
with open(Path(resource_dir, 'col_names.txt')) as cols_fp:
    for line_num, name in enumerate(cols_fp):
        col_names.append(name.rstrip())

data.columns = col_names
```

AFTER APPENDING COLUMN NAMES

frame.offset_shift	frame.time_delta_displayed	radiotap.flags.shortgi	radiotap.rxflags.badplcp	wlan.fc.version	wlan.fc.type	wlan.fc.frag
0.0	0.0	0	0	0	0	0
0.0	0.024271	0	0	0	0	0
0.0	0.001631	0	0	0	0	0
0.0	0.055325	0	0	0	0	0
0.0	0.000415	0	0	0	2	0
0.0	5e-06	0	0	0	1	0
0.0	0.016692	0	0	0	0	0
0.0	0.000142	0	0	0	1	0
0.0	0.028067	0	0	0	0	0

DROPPED COLUMNS NOT LISTED ON COURSE WEBPAGE

REPLACED '?' WITH NAN VALUES, THEN DROPPED COLUMNS WITH OVER 60% NAN VALUES

- removed 7 columns

```
...

data = data.replace('?', np.nan)

...

# If over 60% of the values in a column is null, remove it
prev_num_cols = len(data.columns)
data.dropna(axis='columns', thresh=len(data.index) * 0.40, inplace=True)
print("Removed " + str(prev_num_cols - len(data.columns)) +
      " columns with all NaN values.")
```

DROP THE COLUMNS THAT HAVE OVER 50% OF ITS VALUES AS CONSTANT

```
for col in data:
    if data[col].nunique() >= (len(data.index) * 0.50):
        cols_to_drop.append(col)

data.drop(columns=cols_to_drop, inplace=True)
```

DROP THE ROWS WITH AT LEAST ONE NAN VALUE IN IT

- ~ 2000 rows

```
data.dropna(inplace=True)
```

OUTPUT THE RELATIVELY CLEAN DATA TO A NEW FILE

```
# Output the minimized and preprocessed dataset to a ZIP file
# (with no index column added)
data.to_csv(
    Path(resource_dir, 'preproc_dataset.zip'),
    sep=',',
    index=False,
    compression='zip')
```

PERFORM MIN-MAX NORMALIZATION ON ATTRIBUTES USED FOR CLASSIFICATION (RANGE 0-1)

```
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x)))  }

...

wifiLog2$wlan.fc.type=normalize(as.numeric(wifiLog2$wlan.fc.type))

wifiLog2$frame.time_delta_displayed=normalize(as.numeric(
    wifiLog2$frame.time_delta_displayed
))

wifiLog2$wlan.duration=normalize(as.numeric(wifiLog2$wlan.duration))
View(wifiLog2)
```

NORMALIZATION OUTPUT

KNN-CONTINUOUS.R* | wifiLog | wifiLog2 | wifiLog2 | train | test

Filter

	wlan.fc.type	frame.time_delta_displayed	wlan.duration	class
1	0.0	0.000000e+00	0.00310559	normal
2	0.0	6.929365e-02	0.00310559	normal
3	0.0	4.656501e-03	0.00310559	normal
4	0.0	1.579527e-01	0.00310559	normal
5	1.0	1.184824e-03	0.64285714	normal
6	0.5	1.427499e-05	0.00310559	normal
7	0.0	4.765562e-02	0.00310559	normal
8	0.5	4.054096e-04	0.00310559	normal
9	0.0	8.013122e-02	0.00310559	normal
10	0.0	5.141851e-03	0.00310559	normal
11	0.0	3.567034e-02	0.00310559	normal
12	1.0	6.566494e-05	0.64285714	normal
13	0.5	1.084899e-04	0.00310559	normal
14	0.0	1.219912e-01	0.00310559	normal
15	0.0	7.223715e-02	0.00310559	normal
16	1.0	1.800932e-02	0.64285714	normal
17	0.5	1.427499e-05	0.00310559	normal
18	1.0	6.480844e-03	0.64285714	normal

FEATURE SELECTION

WE ATTEMPTED PCA, BUT RAN OUT OF MEMORY

…even on CSU's Big Data Servers

WE EXAMINED DISTINCT VALUES IN REMAINING COLUMNS, AND CHOSE THOSE WITH MORE DISTINCT VALUES FOR THE NORMAL CLASS VALUE THAN THE ATTACK CLASS VALUES

USING A LITTLE SQL MAGIC...

```
select count(DISTINCT(wlan_fc_moredata))
  from AWID_REMOVED_NULL where class='normal'

select count(DISTINCT(wlan_fc_moredata))
  from AWID_REMOVED_NULL where class='arp'

select count(DISTINCT(wlan_fc_moredata))
  from AWID_REMOVED_NULL where class='amok'

select count(DISTINCT(wlan_fc_moredata))
  from AWID_REMOVED_NULL where class='authentication_request'

select count(DISTINCT(wlan_fc_moredata))
  from AWID_REMOVED_NULL where class='deauthentication'

select wlan_fc_moredata
  from AWID_REMOVED_NULL where class='normal'
```

THEN, CHOSE THE FOLLOWING 3 COLUMNS FOR OUR ANALYSIS:

wlan.fc.type
frame.time_delta_displayed
wlan.duration

CLASSIFICATION

ISOLATED THE ATTACK TYPES

```
...

ATTACKTYPE<-"amok"

# Keep only the target class and the normal packets
wifiLog2<-wifiLog2[wifiLog2$class=="normal" | wifiLog2$class==ATTACKTYPE, ]

wifiLog2$class<-as.character(wifiLog2$class)
wifiLog2$class[wifiLog2$class=="normal"]<-as.character("0")
wifiLog2$class[wifiLog2$class==ATTACKTYPE]<-as.character("1")
wifiLog2$class<-as.factor(wifiLog2$class)

...
```

SEPARATE FILES TO HANDLE EACH ATTACK TYPE

```
KNN-CONTINUOUS-AMOK.R
KNN-CONTINUOUS-ARP.R
KNN-CONTINUOUS-AUTHENTICATION_REQUEST.R
KNN-CONTINUOUS-DEAUTHENTICATION.R
```

PARTITIONED DATASET INTO 66.6% TRAINING DATA AND 33.3% TEST DATA

```
smp_size <- floor(0.66 * nrow(wifiLog2))
```

PERFORMED SMOTE ON TRAINING DATA

- To create synthetic tuples of attack types

```
f<-formula("class~wlan.fc.type+frame.time_delta_displayed+wlan.duration")
train_smote<-SMOTE(f,train,perc.over=150,perc.under=90,k=3)
View(train_smote)
```

K-NEAREST NEIGHBOR CLASSIFIER TO TRAIN MODEL FOR EACH SPECIFIC ATTACK TYPE

```
m<-kNN(f,train_smote,test_oversamp,norm=FALSE,k=5)
```

MADE PREDICTIONS USING THE MODEL ON THE TEST DATASET

PARAMETER SELECTION/INTERPRETATION

RECALL - "COMPLETENESS – WHAT % OF POSITIVE TUPLES DID THE CLASSIFIER LABEL AS POSITIVE?"

$$recall = \frac{TP}{TP + FN}$$

PRECISION - "EXACTNESS – WHAT % OF TUPLES THAT THE CLASSIFIER LABELED AS POSITIVE ARE ACTUALLY POSITIVE"

$$precision = \frac{TP}{TP + FP}$$

RECALL AND PRECISION ARE INVERSELY RELATED MEASURES, MEANING AS PRECISION INCREASES, RECALL DECREASES.

ACCURACY AND RECALL ARE INVERSELY RELATED IN OUR CASE (FOR A MAJORITY OF OUR DATA)

RESULTS

- Performed multiple tests for each attack

ARP (ADDRESS RESOLUTION PROTOCOL) (TEST 1)

ARP (TEST 1) KNN PARAMETERS

- Smote.k = 3
- knn.k = 5
- smote.perc.over = 150
- smote.perc.under = 90

ARP (TEST 1) - CONFUSION MATRIX

- N = 576,582

	Predicted: NO	Predicted: YES	Total
Actual: NO	552,958	1,731	554,689
Actual: YES	4	21,889	21,893
Total	552,962	23,620	

ARP (TEST 1) - ANOMALY DETECTION METRICS

False Positives	1,731
True Positives	21,889
True Negatives	552,958
False Negatives	4

ARP (TEST 1) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	99.6990%
Error Rate	0.3009%
Sensitivity	92.6714%
Specificity	99.9992%
Precision	92.6714%
Recall	99.9817%

ONLY ONE SET OF RESULTS WITH ARP

- Too many errors using other settings
- Difficult to improve on already extremely good results

AMOK (TEST 1)

AMOK (TEST 1) KNN PARAMETERS

- Smote.k = 3
- knn.k = 5
- smote.perc.over = 150
- smote.perc.under = 90

AMOK (TEST 1) - CONFUSION MATRIX

- N = 565,216

	Predicted: NO	Predicted: YES	Total
Actual: NO	511,451	42,928	554,379
Actual: YES	562	10,275	10,837
Total	512,013	53,203	

AMOK (TEST 1) - ANOMALY DETECTION METRICS

False Positives	42,928
True Positives	10,275
True Negatives	511,451
False Negatives	562

AMOK (TEST 1) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	92.3056%
Error Rate	7.6944%
Sensitivity	19.3128%
Specificity	99.8902%
Precision	19.3128%
Recall	94.8140%

AMOK (TEST 2)

AMOK (TEST 2) KNN PARAMETERS

- smote.k = 1
- knn.k = 1
- smote.perc.over = 120
- smote.perc.under = 200

AMOK (TEST 2) - CONFUSION MATRIX

- N = 565,216

	Predicted: NO	Predicted: YES	Total
Actual: NO	529,906	24,473	554,379
Actual: YES	1099	9,738	10,837
Total	531,005	34,211	

AMOK (TEST 2) - ANOMALY DETECTION METRICS

False Positives	24,473
True Positives	9,738
True Negatives	529,906
False Negatives	1099

AMOK (TEST 2) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	95.4757%
Error Rate	4.5242%
Sensitivity	2.8464%
Specificity	99.7930%
Precision	28.4645%
Recall	89.8588%

DEAUTHENTICATION (TEST 1)

DEAUTHENTICATION (TEST 1) KNN PARAMETERS

- Smote.k = 3
- knn.k = 5
- smote.perc.over = 150
- smote.perc.under = 90

DEAUTHENTICATION (TEST 1) - CONFUSION MATRIX

- N = 558,167

	Predicted: NO	Predicted: YES	Total
Actual: NO	512,542	42,022	554,564
Actual: YES	95	3,508	3,603
Total	512,637	45,530	

DEAUTHENTICATION (TEST 1) - ANOMALY DETECTION METRICS

False Positives	42,022
True Positives	3,508
True Negatives	512,542
False Negatives	95

DEAUTHENTICATION (TEST 1) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	92.4544%
Error Rate	7.5455%
Sensitivity	7.7048%
Specificity	99.9814%
Precision	7.7048%
Recall	97.3633%

DEAUTHENTICATION (TEST 2)

DEAUTHENTICATION (TEST 2) KNN PARAMETERS

- smote.k = 1
- knn.k = 1
- smote.perc.over = 90
- smote.perc.under = 400

DEAUTHENTICATION (TEST 2) - CONFUSION MATRIX

- N = 558,167

	Predicted: NO	Predicted: YES	Total
Actual: NO	527,780	26,784	554,564
Actual: YES	379	3,224	3,603
Total	528,159	30,008	

DEAUTHENTICATION (TEST 2) - ANOMALY DETECTION METRICS

False Positives	26,784
True Positives	3,224
True Negatives	527,780
False Negatives	379

DEAUTHENTICATION (TEST 2) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	95.1335%
Error Rate	4.8664%
Sensitivity	10.7438%
Specificity	99.9282%
Precision	10.7438%
Recall	89.4809%

AUTHENTICATION REQUEST (TEST 1)

AUTHENTICATION REQUEST (TEST 1) KNN PARAMETERS

- Smote.k = 3
- knn.k = 5
- smote.perc.over = 150
- smote.perc.under = 90

AUTHENTICATION REQUEST (TEST 1) - ANOMALY DETECTION METRICS

- N = 555,805

	Predicted: NO	Predicted: YES	Total
Actual: NO	513,668	40,945	554,613
Actual: YES	31	1,161	1,192
Total	513,699	42,106	

AUTHENTICATION REQUEST (TEST 1) - ANOMALY DETECTION METRICS

False Positives	40,945
True Positives	1,161
True Negatives	513,668
False Negatives	31

AUTHENTICATION REQUEST (TEST 1) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	92.6276%
Error Rate	7.3723%
Sensitivity	2.7573%
Specificity	99.9939%
Precision	2.7573%
Recall	97.3993%

AUTHENTICATION REQUEST (TEST 2)

AUTHENTICATION REQUEST (TEST 2) KNN PARAMETERS

- Smote.k = 1
- knn.k = 1
- smote.perc.over = 100
- smote.perc.under = 300

AUTHENTICATION REQUEST (TEST 2) - ANOMALY DETECTION METRICS

- N = 555,805

	Predicted: NO	Predicted: YES	Total
Actual: NO	540,840	13,773	554,613
Actual: YES	152	1,040	1,192
Total	540,992	14,813	

AUTHENTICATION REQUEST (TEST 2) - ANOMALY DETECTION METRICS

False Positives	13,773
True Positives	1,040
True Negatives	540,840
False Negatives	152

AUTHENTICATION REQUEST (TEST 2) - ANOMALY DETECTION METRICS (CONTD.)

Accuracy	97.4946%
Error Rate	2.5053%
Sensitivity	7.0208%
Specificity	99.9719%
Precision	7.0208%
Recall	87.2483%

SOURCES

Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset
https://en.wikipedia.org/wiki/Address_Resolution_Protocol

THANK YOU