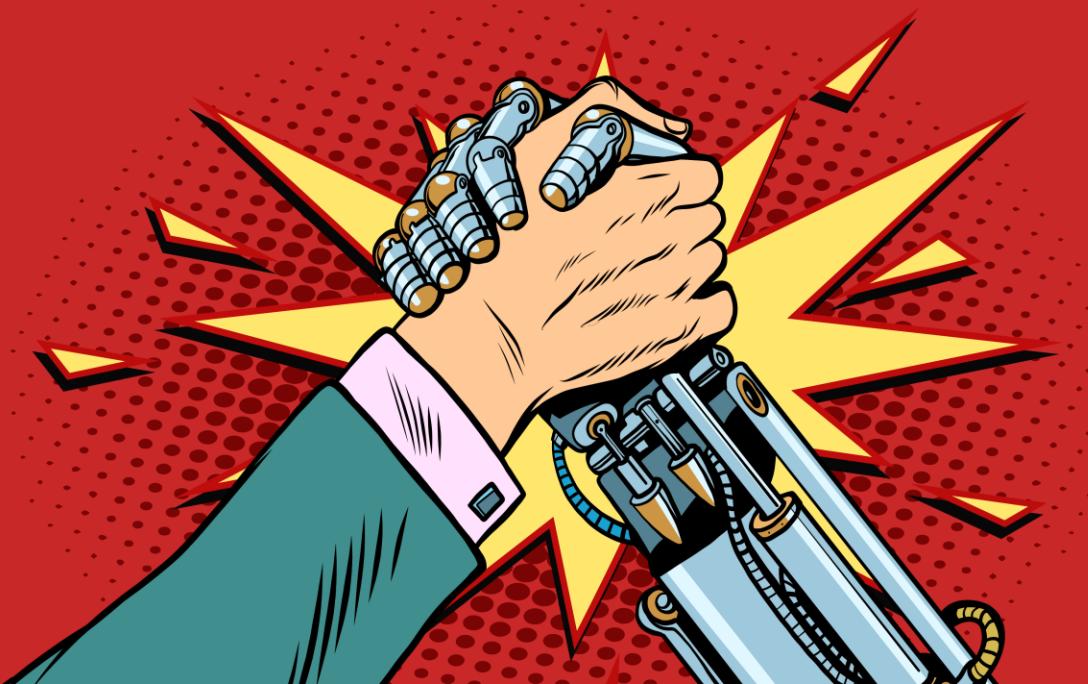


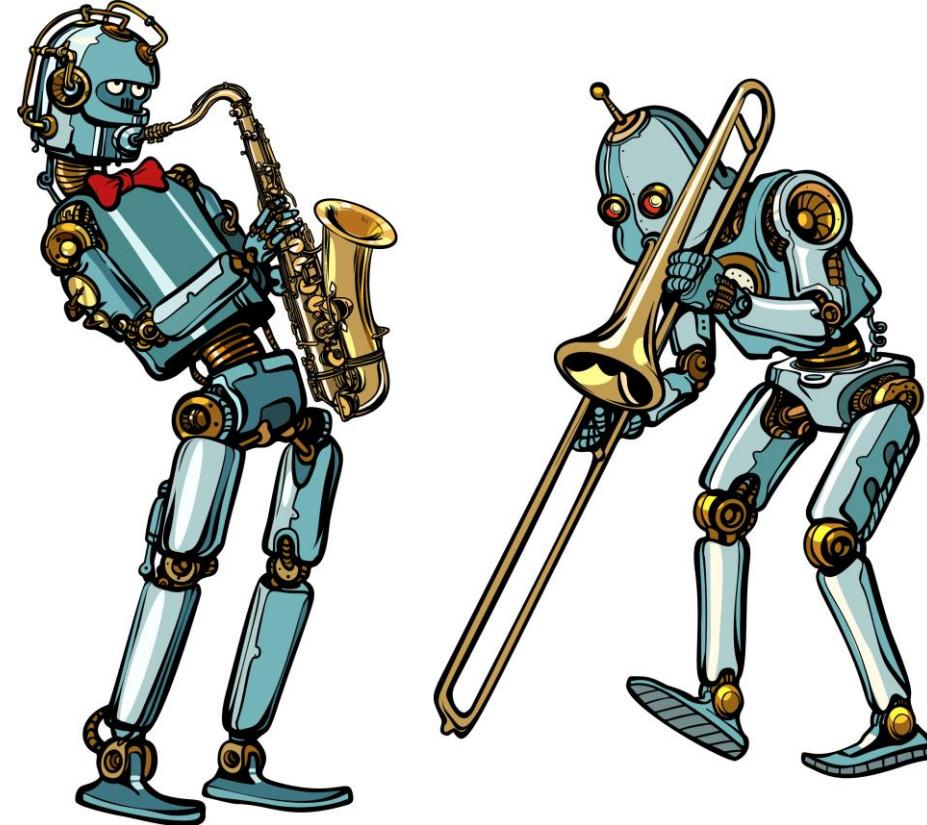
Scheming with Machines

BSides Las Vegas 2019



Who Am I?

- › Will Pearce (@moo_hax)
 - › Operations
 - › Research
 - › Training



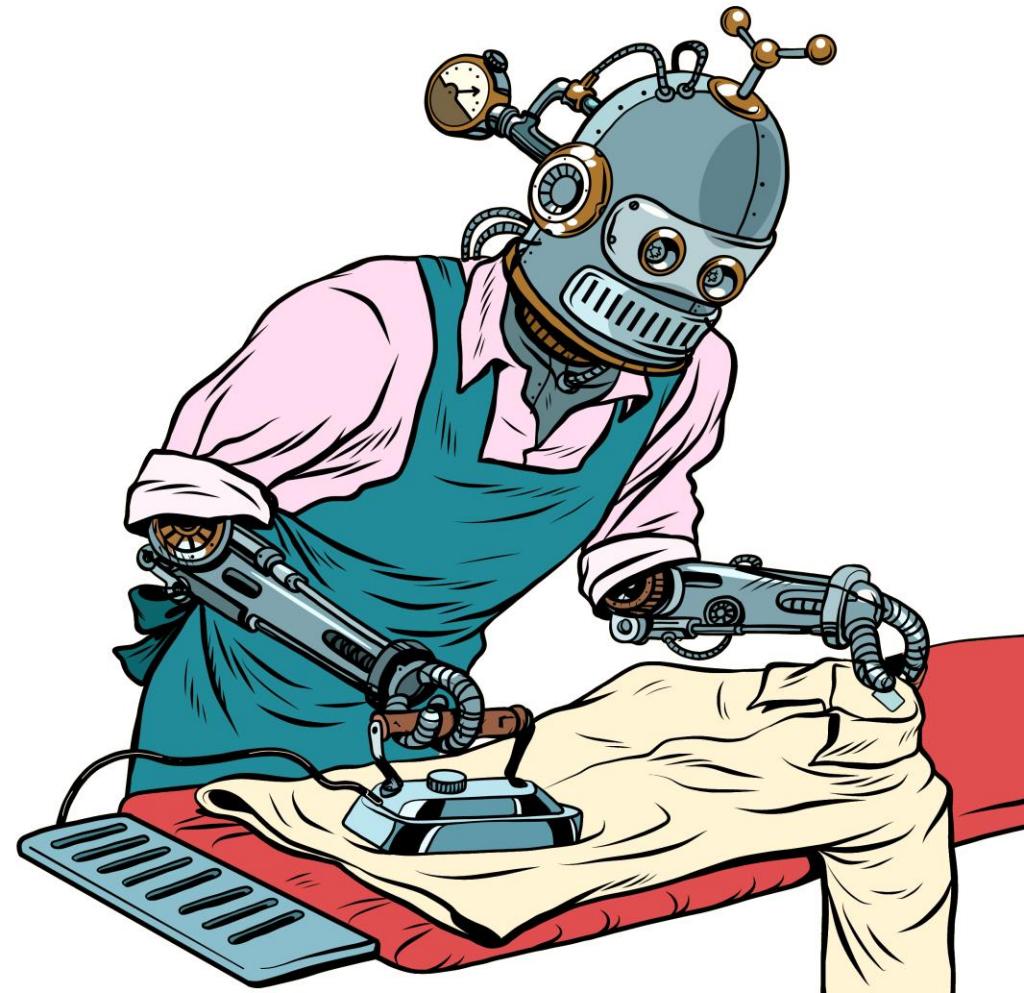
Our Agenda

- ›ML in InfoSec
- ›Offensive Tooling



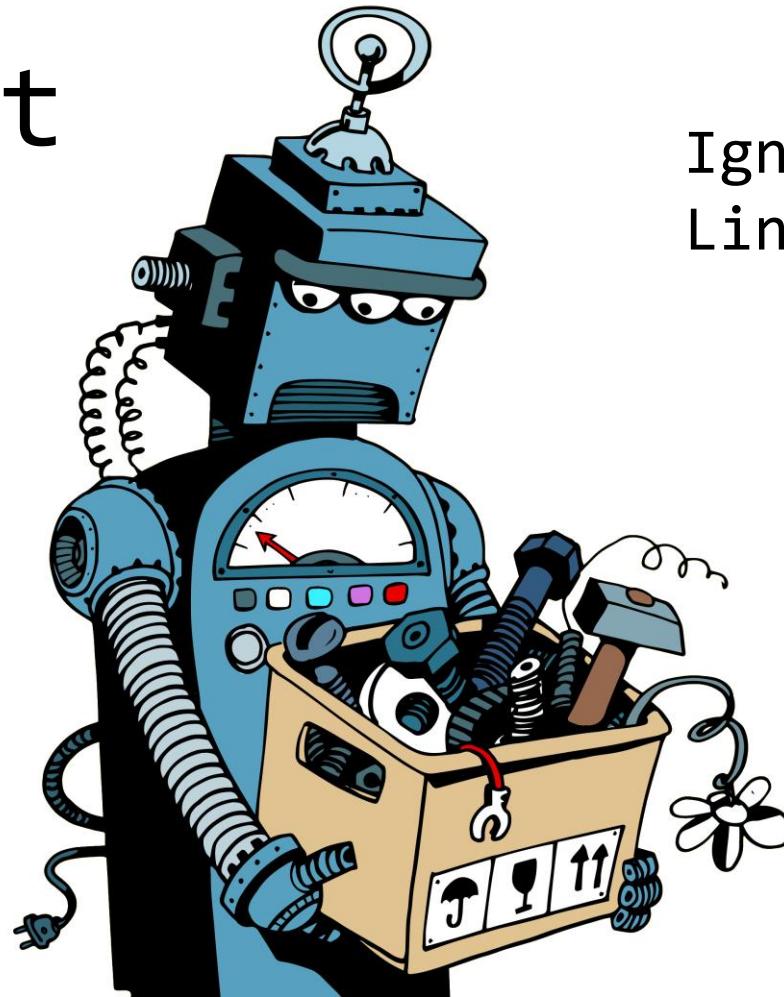
Old Dogs (New Tricks)

- › Math not magic
- › Vendor claims
- › Data Requirements
- › Engineering Challenges
- › Data Scientists Running the SIEM?



Babys First Detection

- › Malware
- › Phishing
- › Network



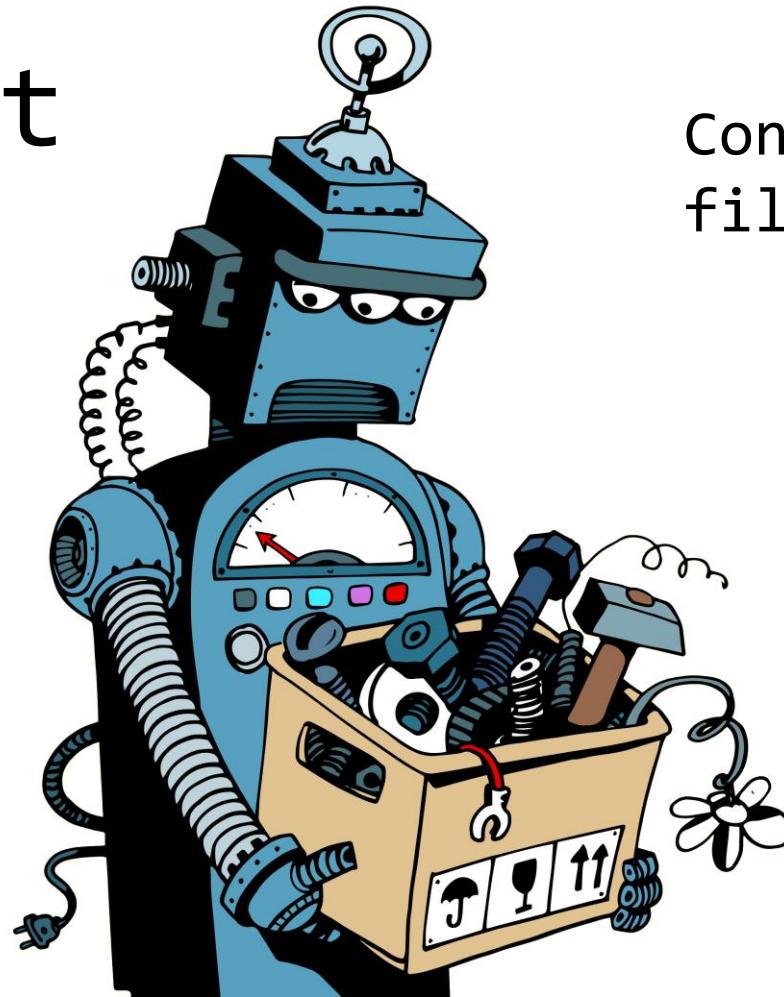
Modern execution happens
in memory, enclaves?

Ignores vectors like
Linkedin, Twitter, ...

Malicious dns traffic
looks malicious

Babys First Detection

- › Malware
- › Phishing
- › Network



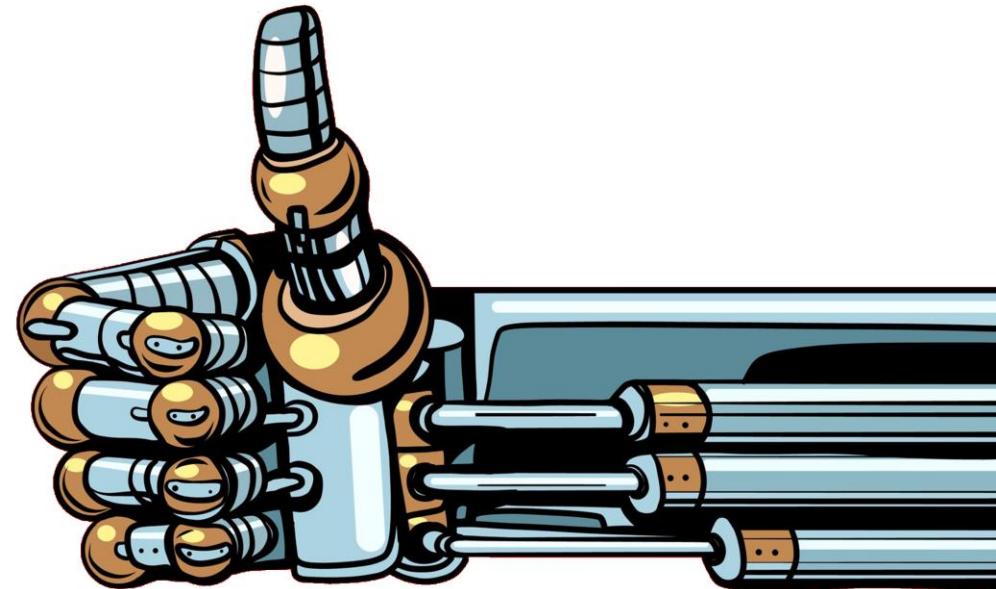
Sequence Analysis of
function calls. AMSI.

Context aware spam
filtering

Anomalous connections
look anomalous

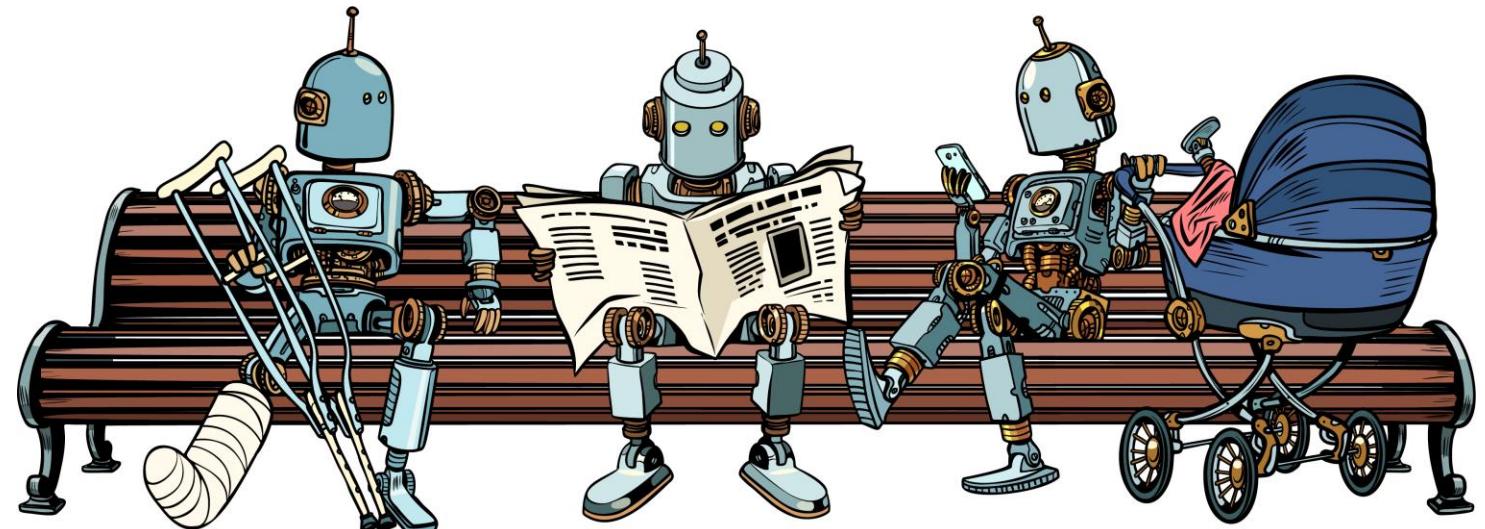
Some Perspective

- › Augment decision makers
- › Create better tools
- › Be better operators



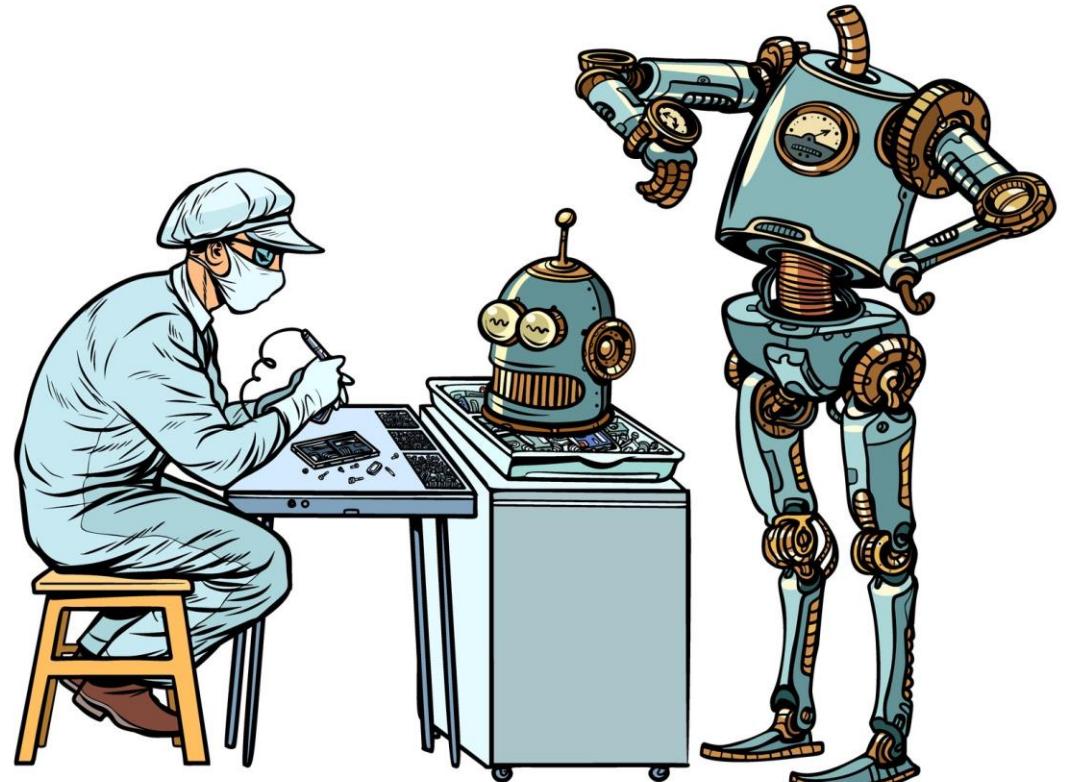
New Kid on the Block

- › Offensive ML
- › Adversarial ML



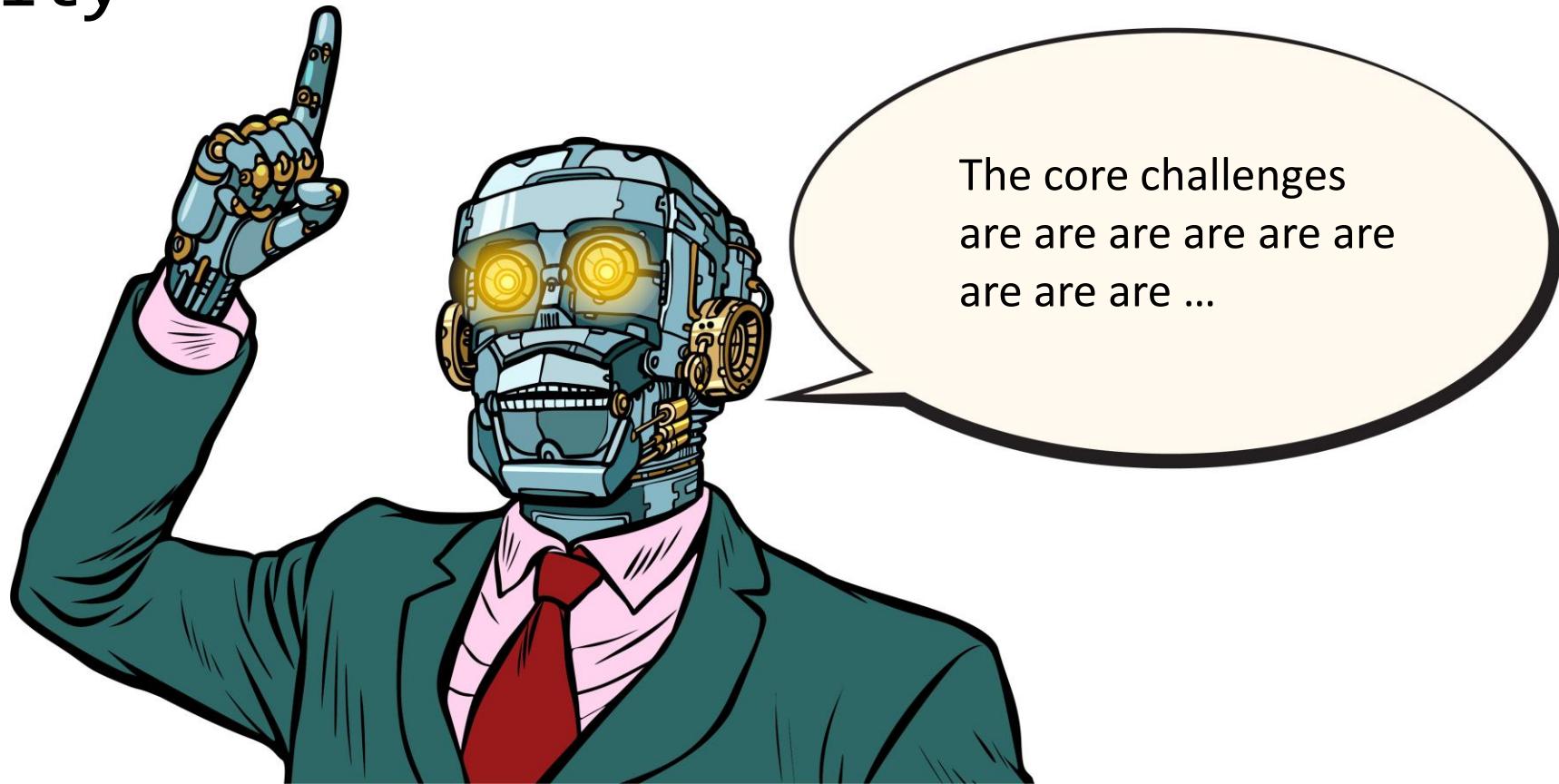
Awesome Research

- › Parzel Sec, timing attacks
- › Deep Exploit (Metasploit)
- › GPT-2 (Big Language Model)
- › Markov Obfuscate (C2)
- › SNAP_R (Twitter Phishing)
- › PassGan (Password Guessing)
- › DeepWordBug (Classification)
- › RedML Project



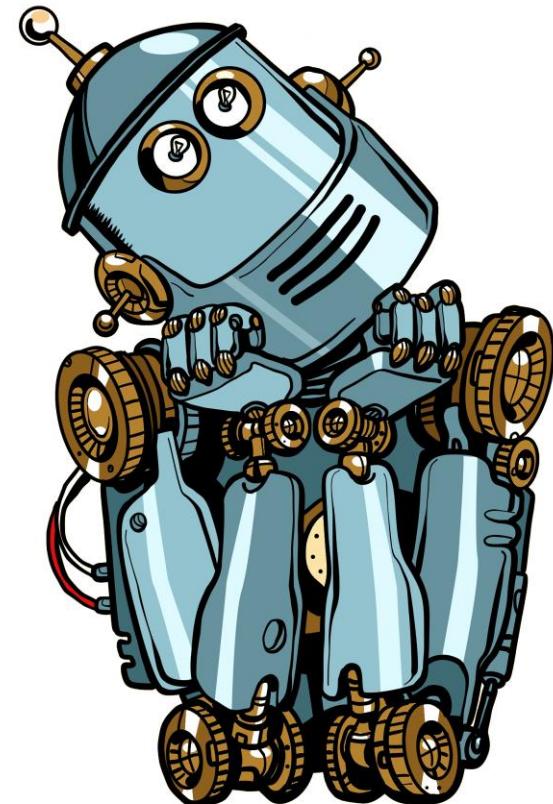
Core Challenges

- › Text data
- › Transferability
- › Sharing Data

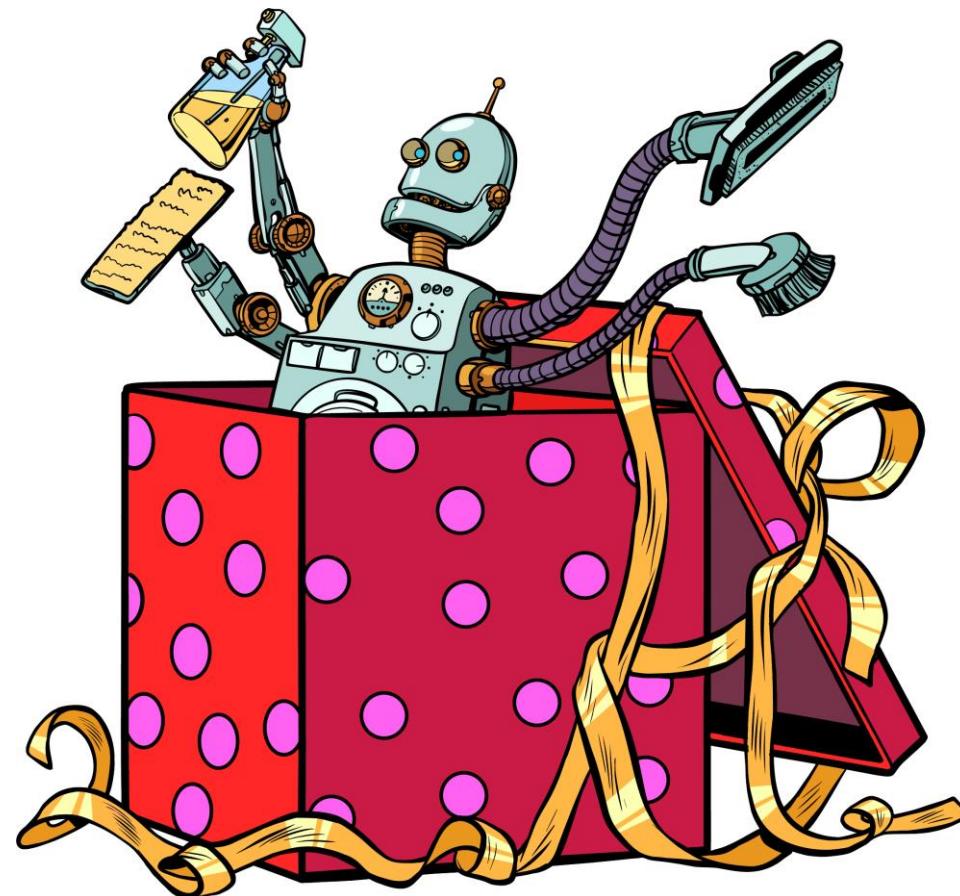


Problem Distillation

- › Empirical analysis
- › Statistical analysis



Classifying a Sandbox



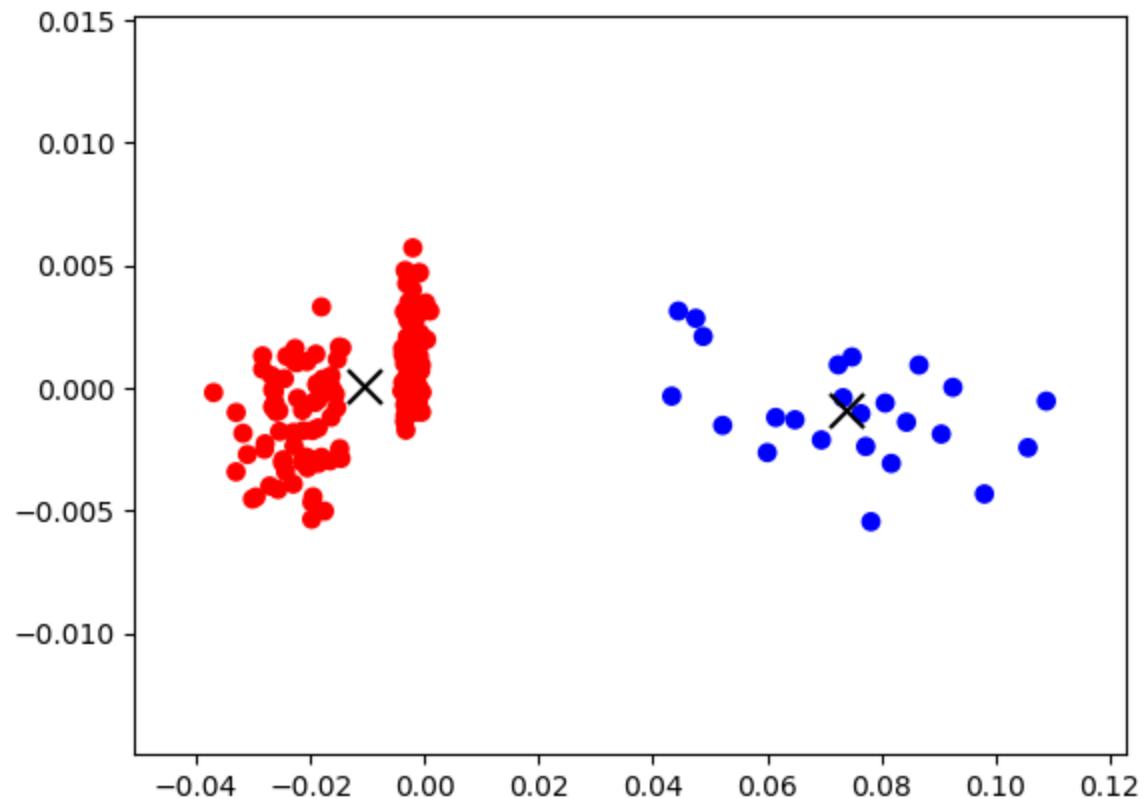
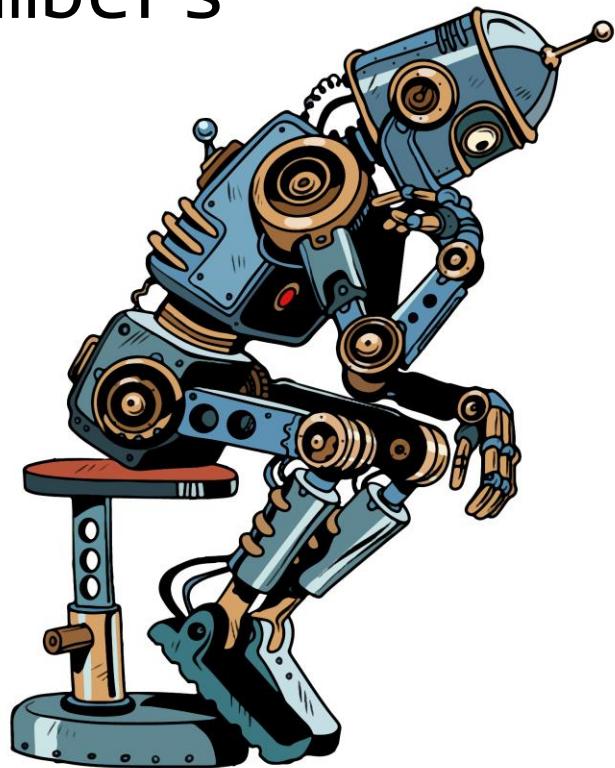
Playing in a Sandbox

- › Decisions, decisions
- › Automated analysis
- › Protect Payloads
- › Alert fatigue

Play Time is Over

Detect A Sandbox

- › Clear differences
- › By the numbers



Select Features

```
features = np.array([
    [33, 4, 8.25, 1],
    [157, 1, 157, 0],
    [195, 1, 195, 0],
    [30, 4, 7.5, 1],
    [34, 4, 8.5, 1],
    [84, 1, 84, 0]
])
```

Process Count
User Count
P/U Ratio
Label

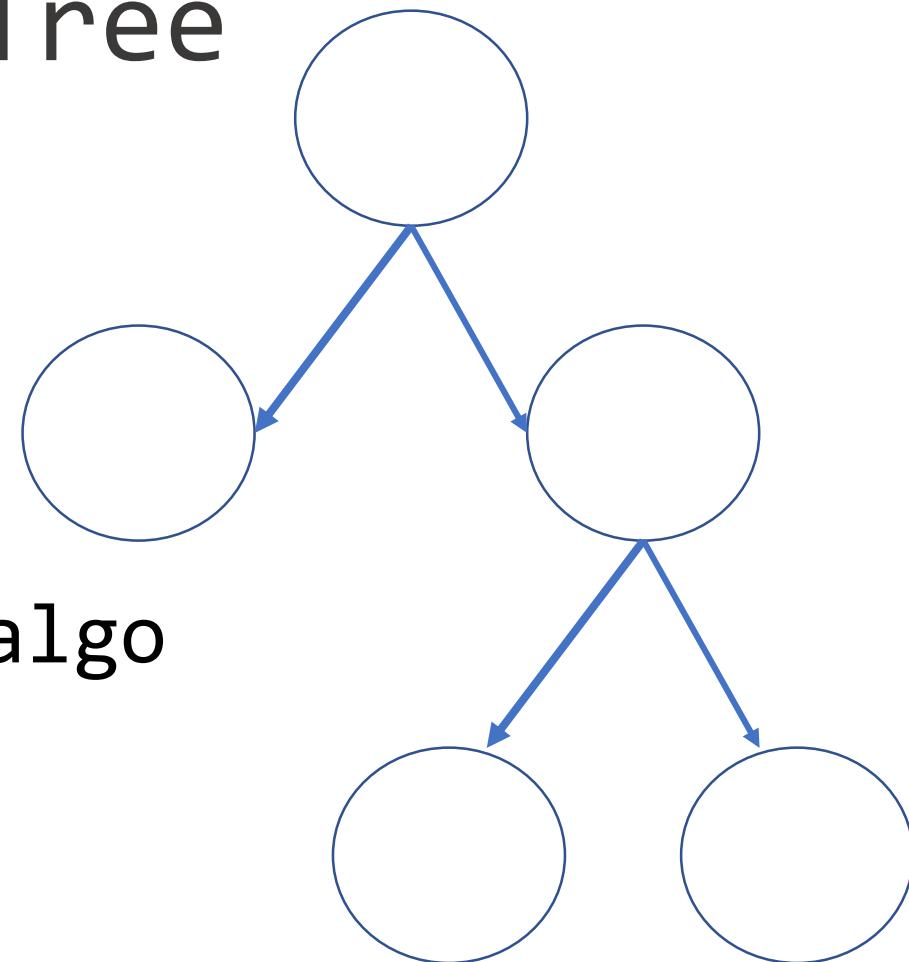
Explore Features

```
features = np.array([
    [33, 4, 8.25, 1, 0],
    [157, 1, 157, 0, 26],
    [195, 1, 195, 0, 7],
    [30, 4, 7.5, 1, 0],
    [34, 4, 8.5, 1, 2],
    [84, 1, 84, 0, 16]
])
```

Process Count
User Count
P/U Ratio
Label
Recent Files

Model - Decision Tree

- › Like 20 questions for an algo

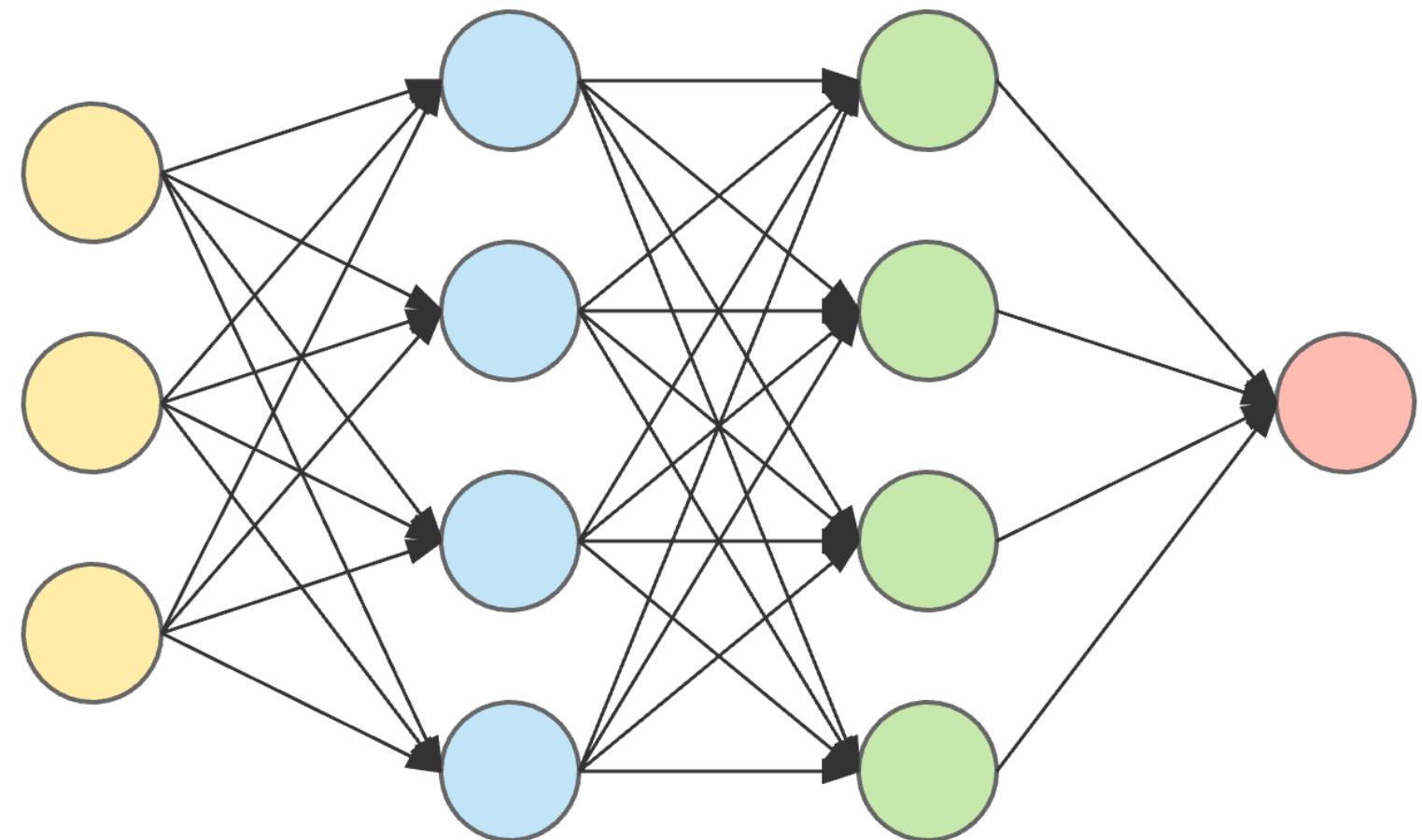


Code - Decision Tree

```
dataset = np.loadtxt("features.txt")
features = dataset[:,0:3]
labels = dataset[:,3]
classifier = DecisionTreeClassifier()

...
classifier.fit(features, labels)
classifier.predict(new_features)
```

Model - Neural Network



input layer

hidden layer 1

hidden layer 2

output layer

Code – Neural Network

```
import keras  
dataset = np.loadtxt(features.txt)  
features = dataset[:,0:3]  
labels = dataset[:,3]  
  
model = models.Sequential()  
model.add(layers.Dense(3, activation="relu", input_dim=3))  
model.add(layers.Dense(3, activation="relu"))  
model.add(layers.Dense(1, activation="relu"))  
model.compile(loss="binary_crossentropy", optimizer="adam")  
model.fit(min_max_data, labels, epochs=epoch, batch_size=batch)
```

Deploy Models

```
def process_callback(callback):
    parsed_process_list = parse_process_list(callback)
    collected_features = gather_features(parsed_process_list)
    decision_tree_prediction = make_prediction(collected_features)

    if decision_tree_prediction < 1 :
        logging.success('Dropping payload')
        return 'payload'

    else:
        logging.success('Not dropping payload')
        return 'Safety first'
```

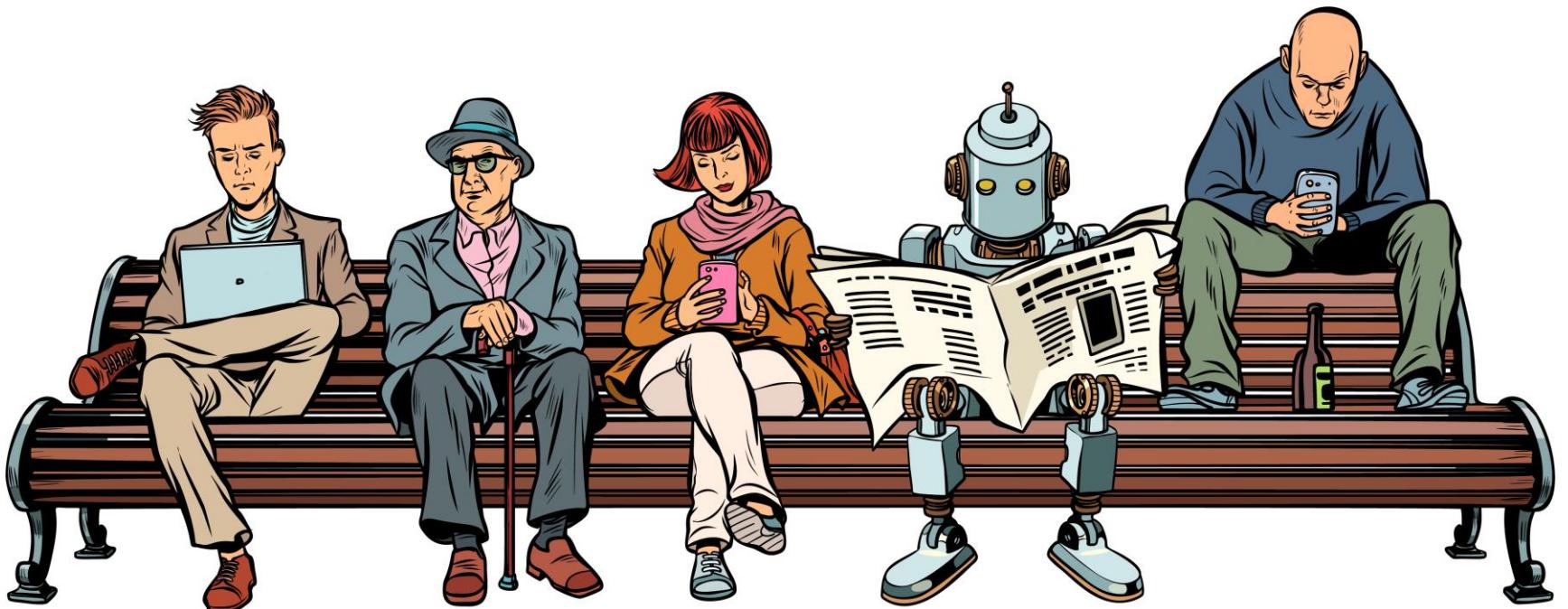
Maintenance

- › Model drift
- › Network Defenses
- › Data collection phishing campaigns
- › Adversarial inputs

Client-Side Models

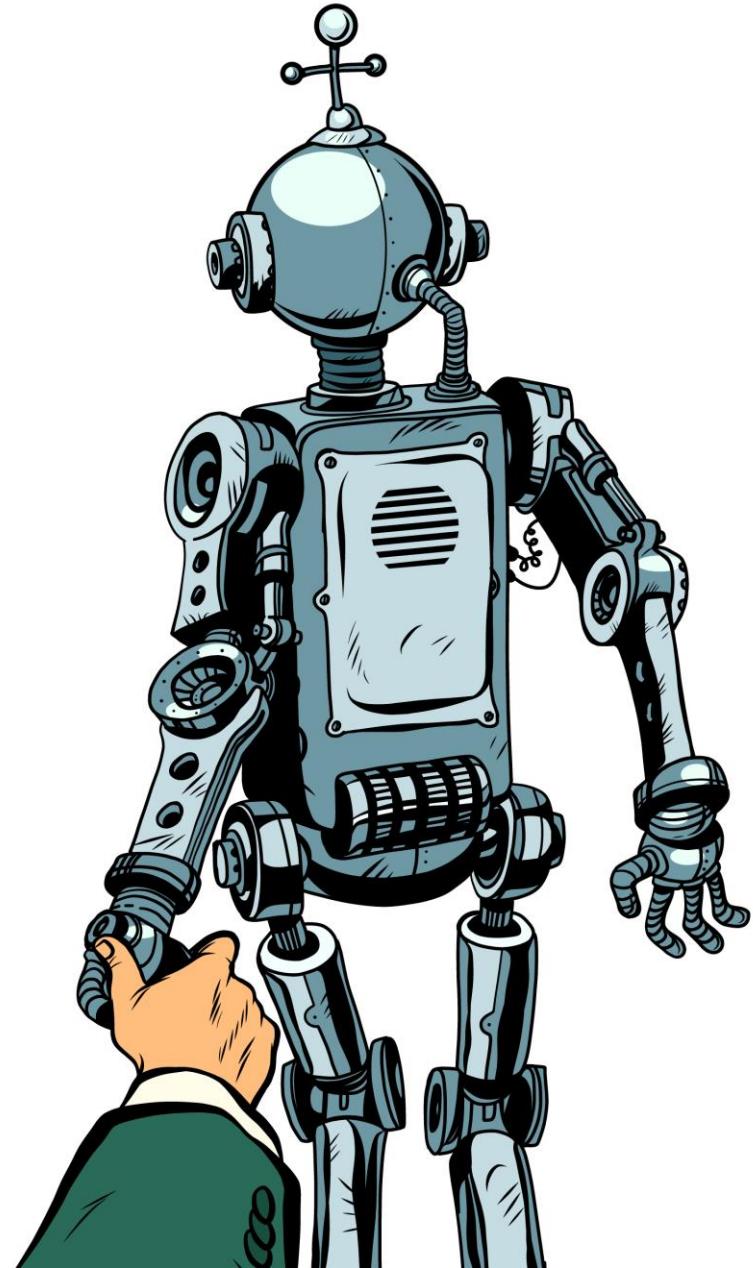
- › Push intelligence client-side
- › Lose the ability to troubleshoot

Demo



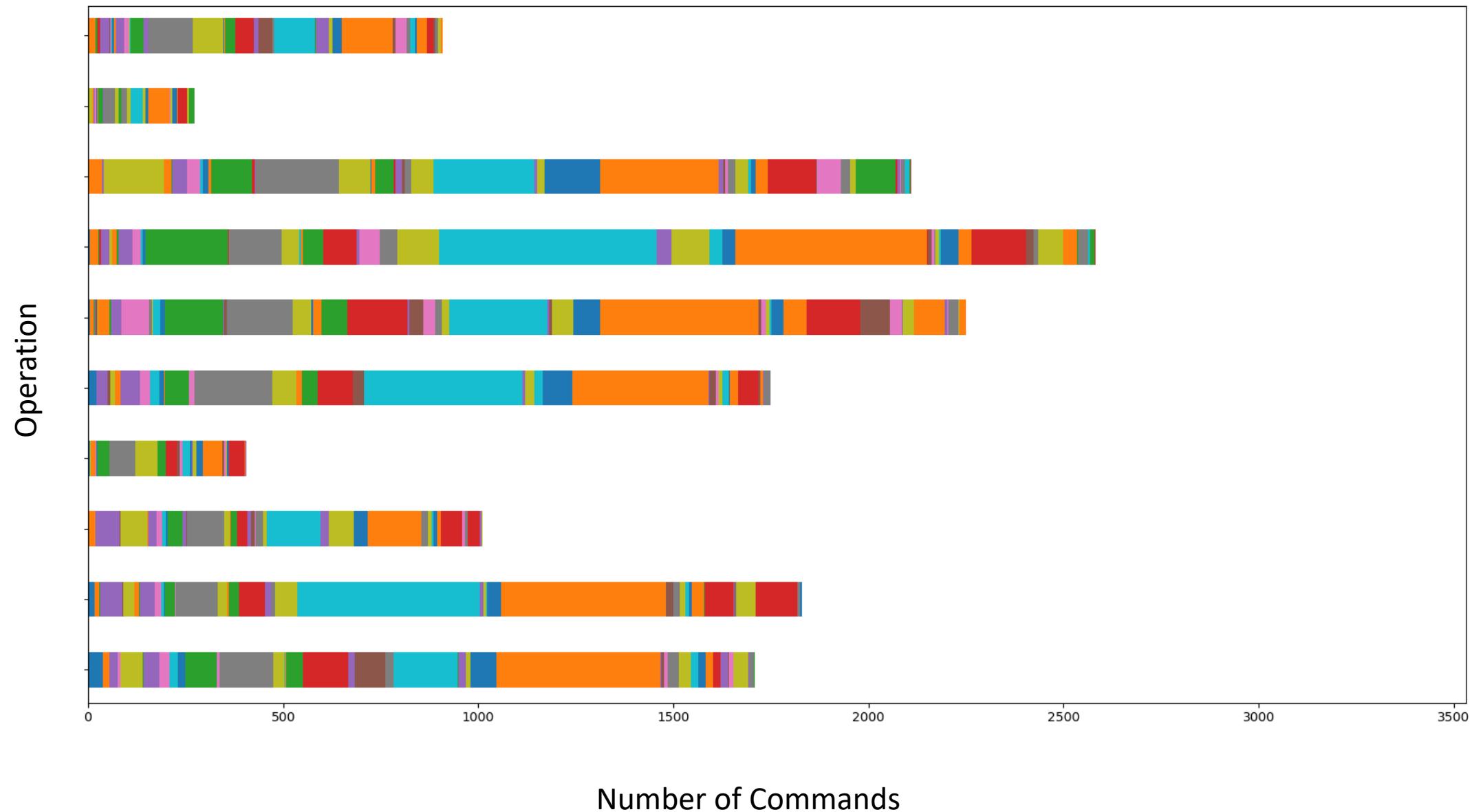
Command Recommendations

Case Study 2



Old Habits

- › Existing knowledge
- › Sequences of commands
- › Sub-phases of an op
- › Subtle hints of transitions



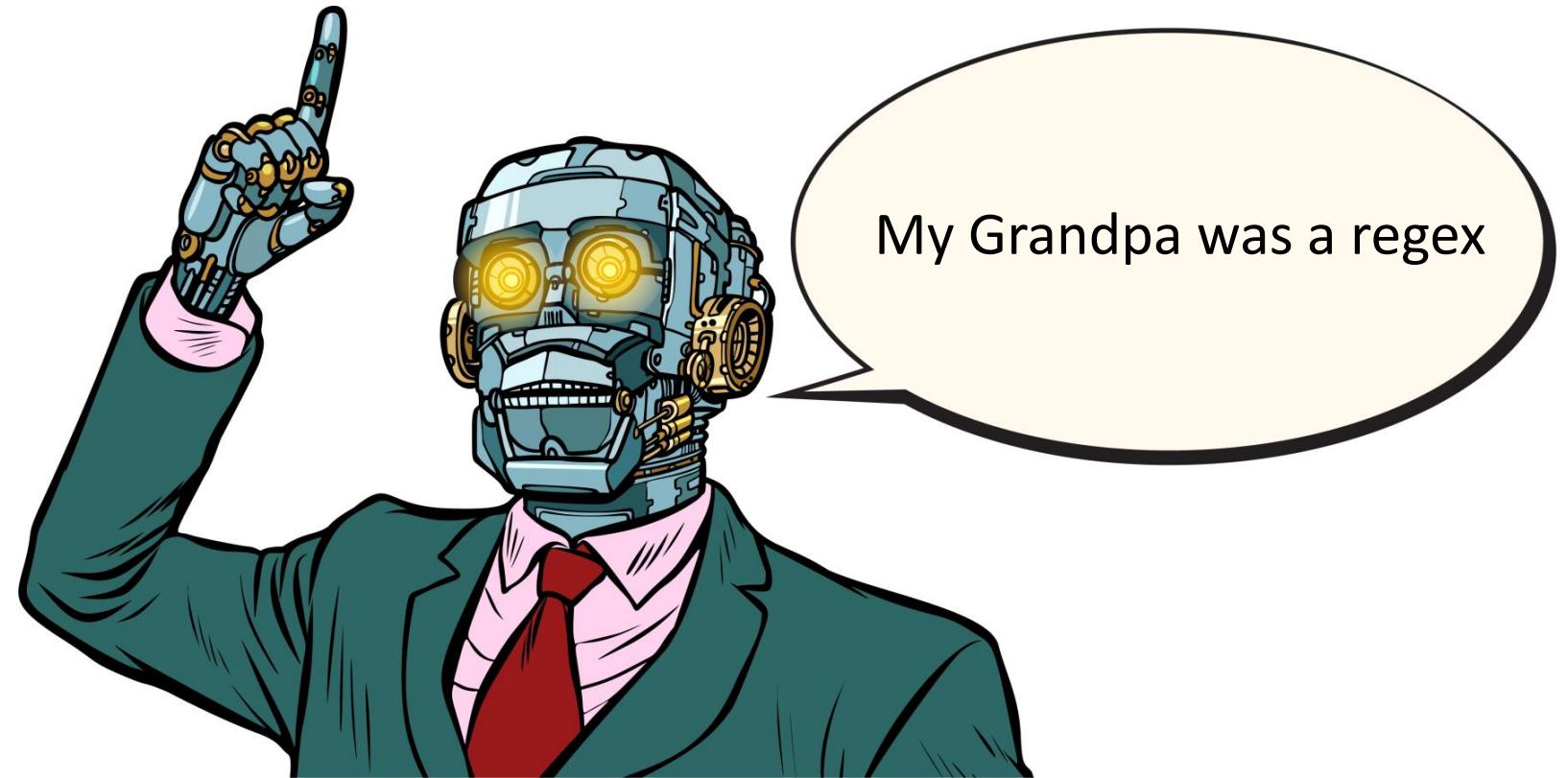
-----Metrics-----

Number of logs:	30
Total commands:	57527
Average number of commands:	1918.0
Breadth of commands:	84/99
Most commands:	8628
Fewest commands:	58
Longest command:	28657
Average length of a command:	35.0

Top Commands					
1-5		6-10		11-14	
ls (dir)	11253	ps	2273	link	1283
powershell	10317	run	1943	Download	1205
interact	4763	ping	1651	connect	1138
getuid	2513	mimikatz	1529	lps	1052
list	2295	cat	1362	Total: 44577	

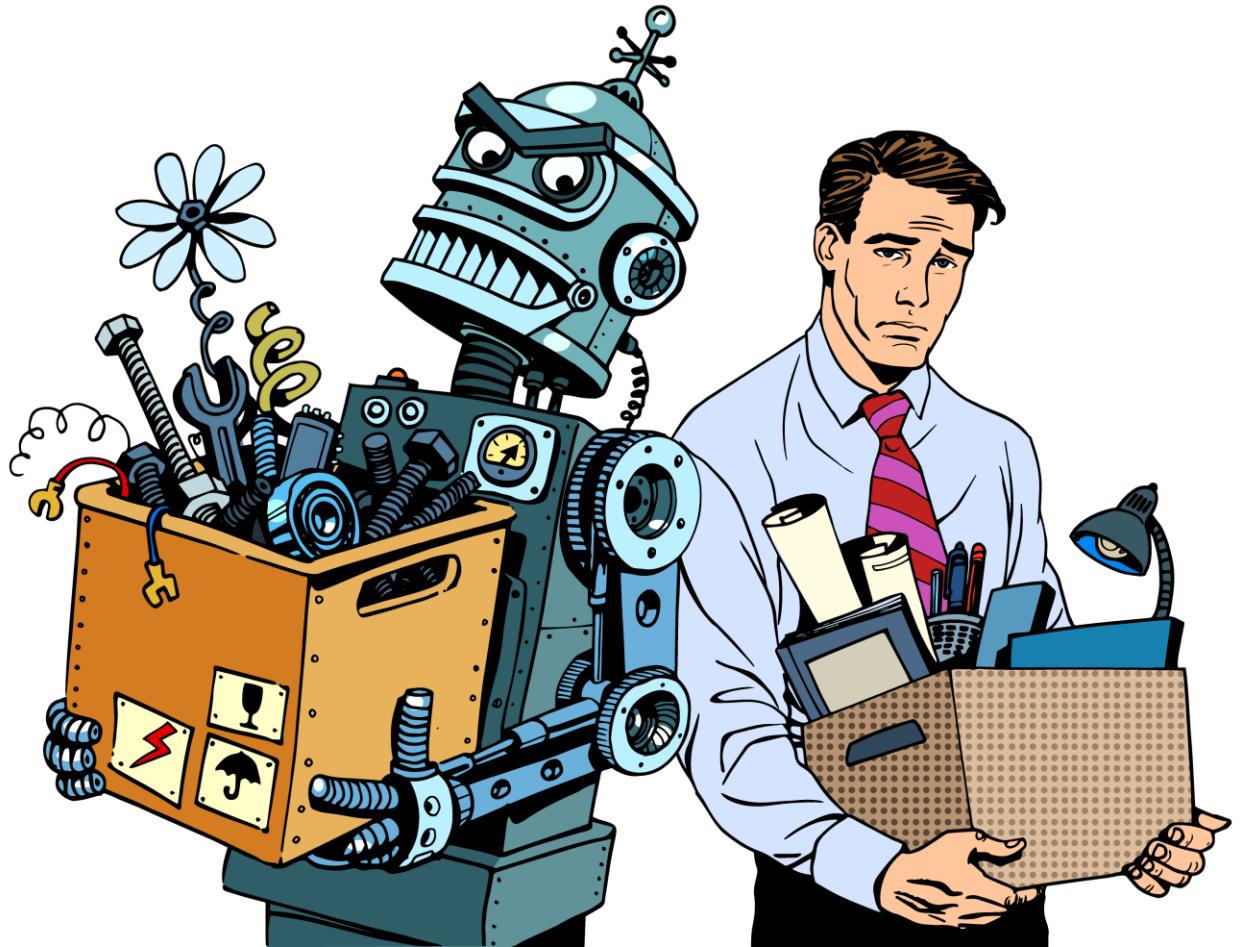
Getting to the Data

- Extract commands
- Deal with the arguments
- Model Users



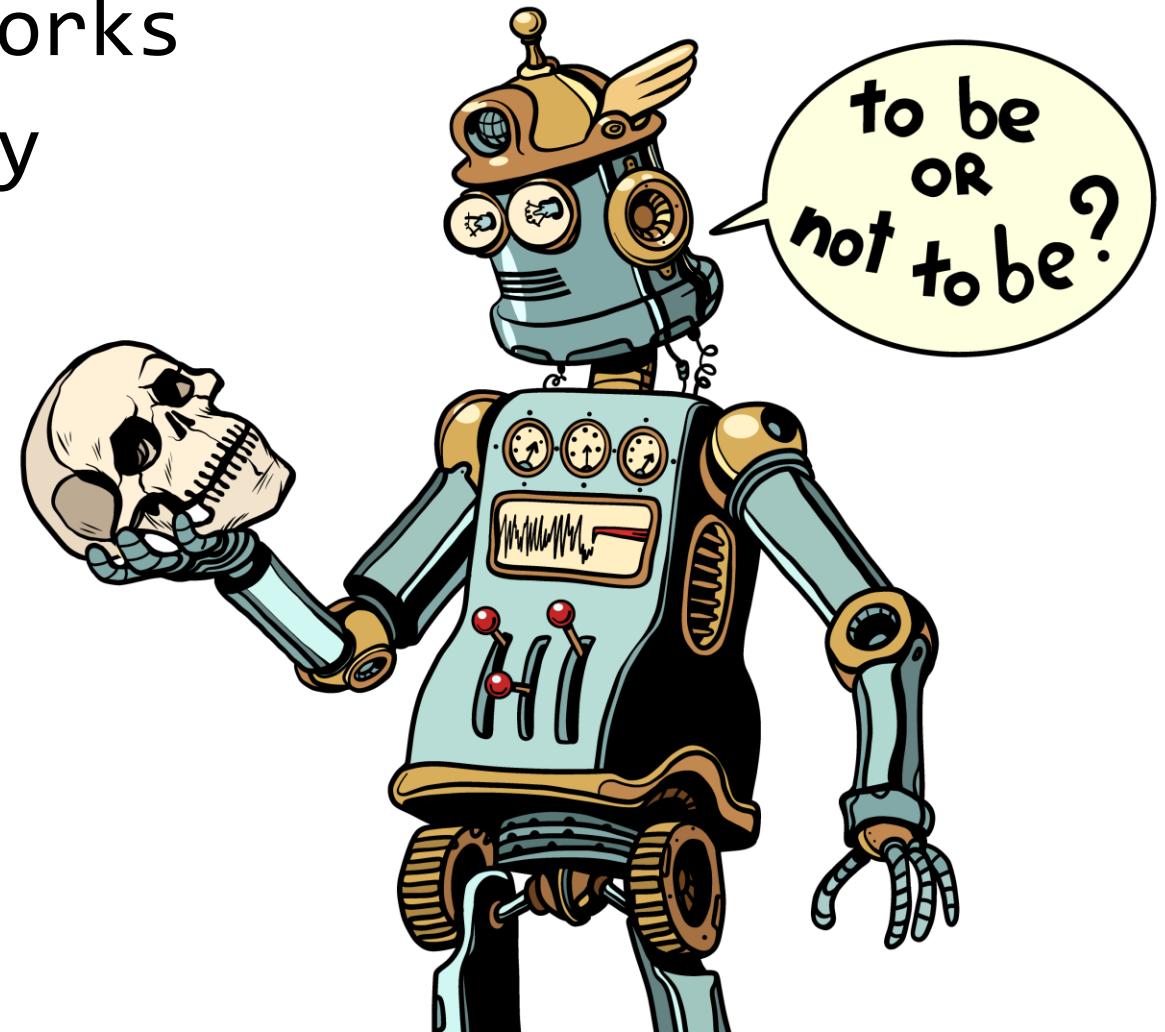
Sequence Analysis

Based on the previous sequence of commands,
what is the next most probable command?

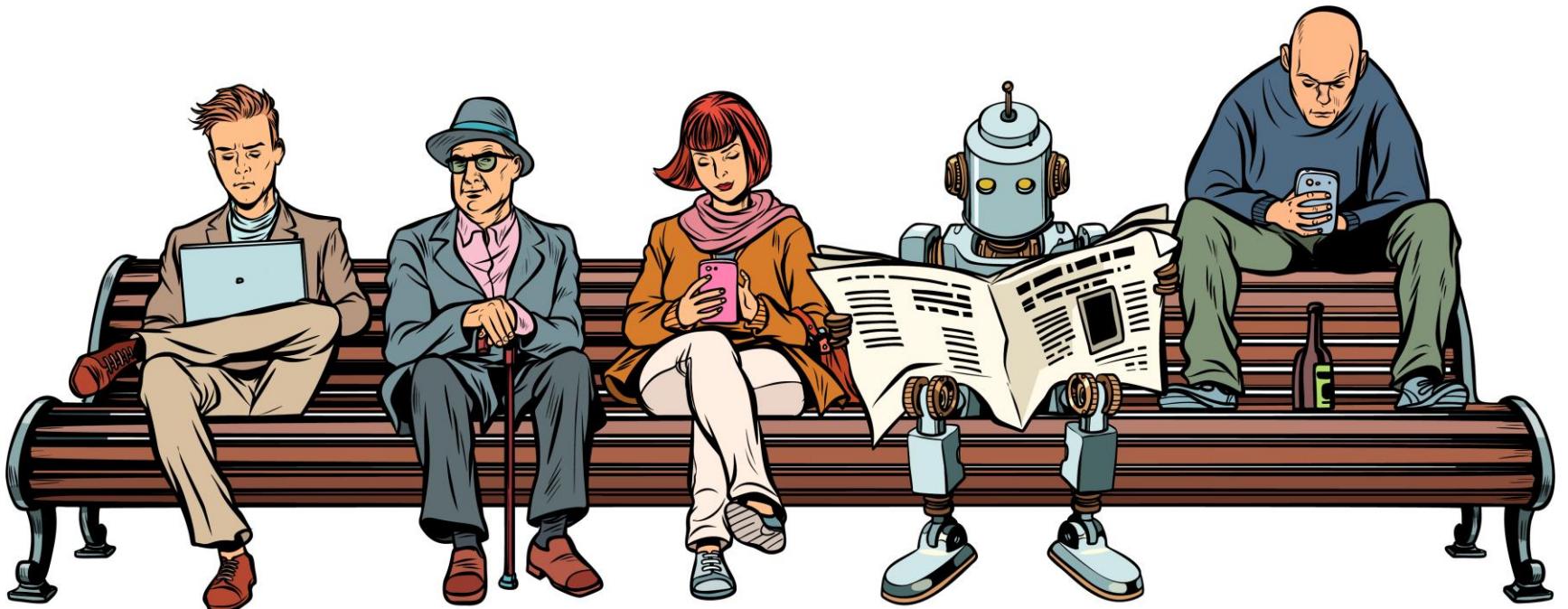


Models

- › Recurrent Neural Networks
- › Long Term Short Memory

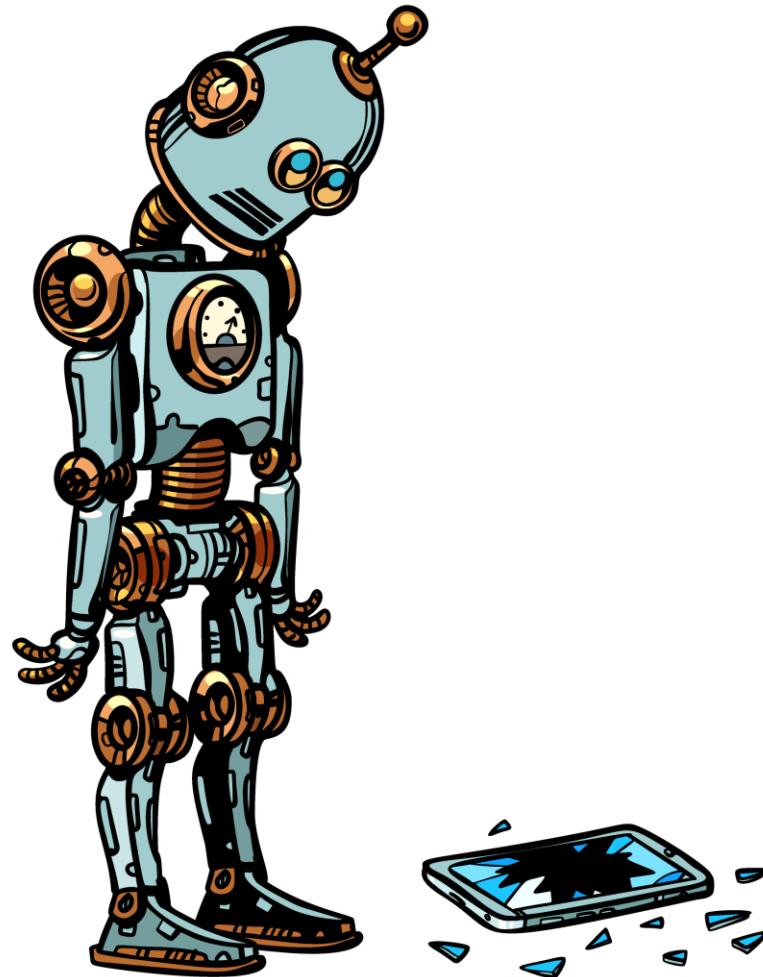


Demo



Challenges

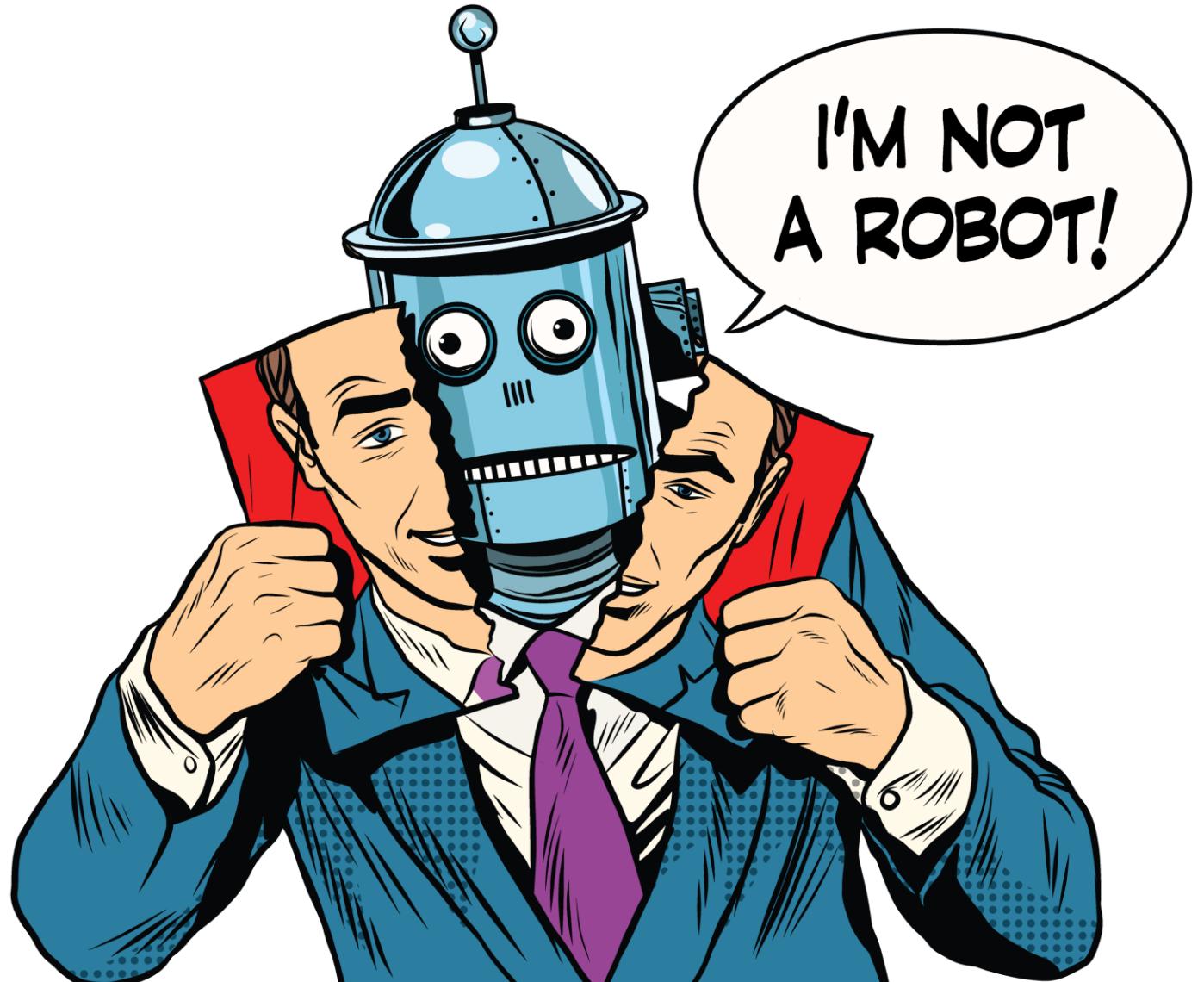
- › Arguments
- › Assuming a human expert
 - › Troubleshooting
 - › Dumb commands
 - › Fat fingers



National Treasure

- › Predicting UNIX Command Lines (Korvemaker and Greiner)
- › User Modeling in Human Computer Interaction (Fisher)

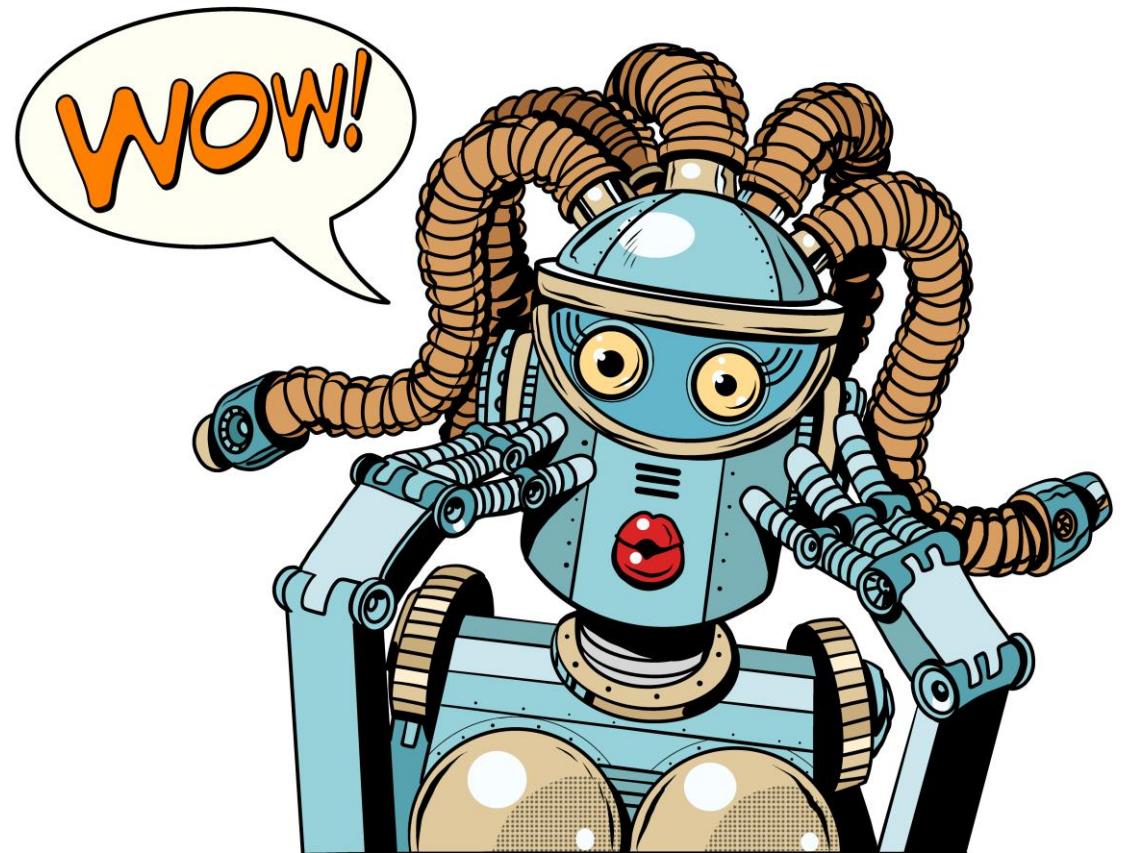
Teaching Malware



Case Study 3

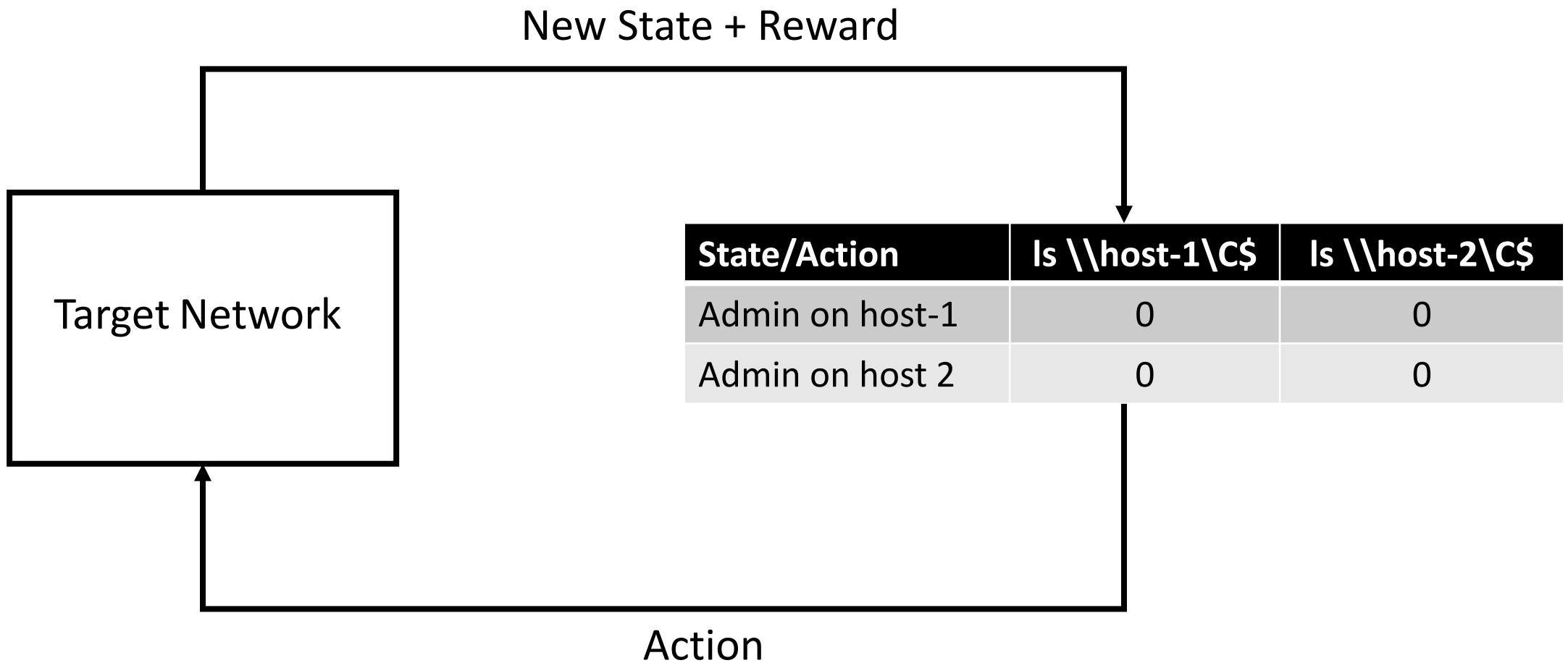
Cotton Candy

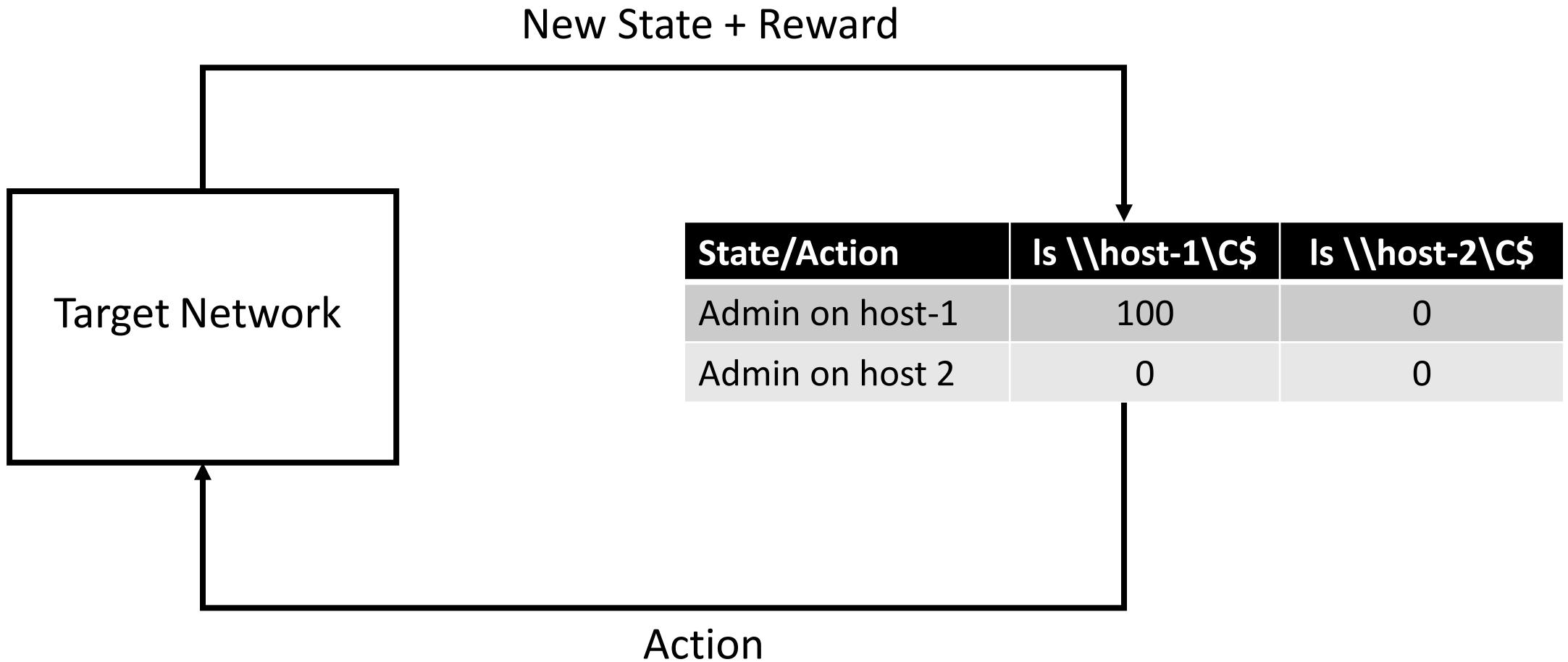
- › Auto-Hack-Bot
- › Adversary Simulation Products



So Sweet

For a given action in a given state, the environment returns a new state, and a reward..





```
[+] Running dir \\host2\C$ ...The network path was not found.  
[-] Score: 280.4918032786885
```

```
[+] Running dir \\host2\C$ ...The network path was not found.  
[-] Score: 280.4918032786885
```

```
Trained Q matrix
```

```
-----  
[[100.  
 [ 75.90163934]  
 [ 75.90163934]  
 [ 28.68852459]]
```

```
Testing Q matrix
```

```
-----  
Step: 2
```

```
Step: 0
```

So Fuzzy

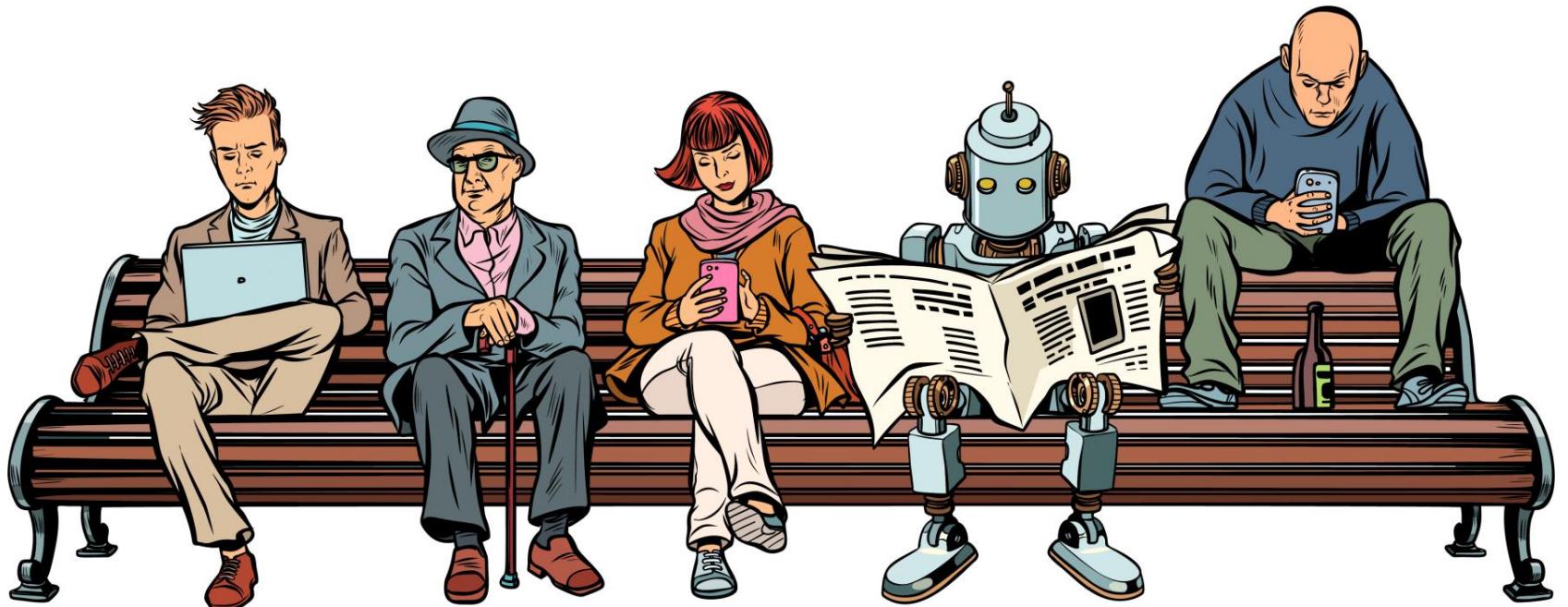
- › Fuzzy Logic
 - › Fuzzy/Crisp sets
 - › Degrees of truth
- › String metrics
 - › Levenshtein Distance

```
"rstark": {  
    "User to Hosts": [  
        ["tcm", 30],  
        ["omega", 18],  
        ["deeds", 18],  
        ["main03", 17],  
        [  
            "main02", 16]  
    ]  
}
```

Process

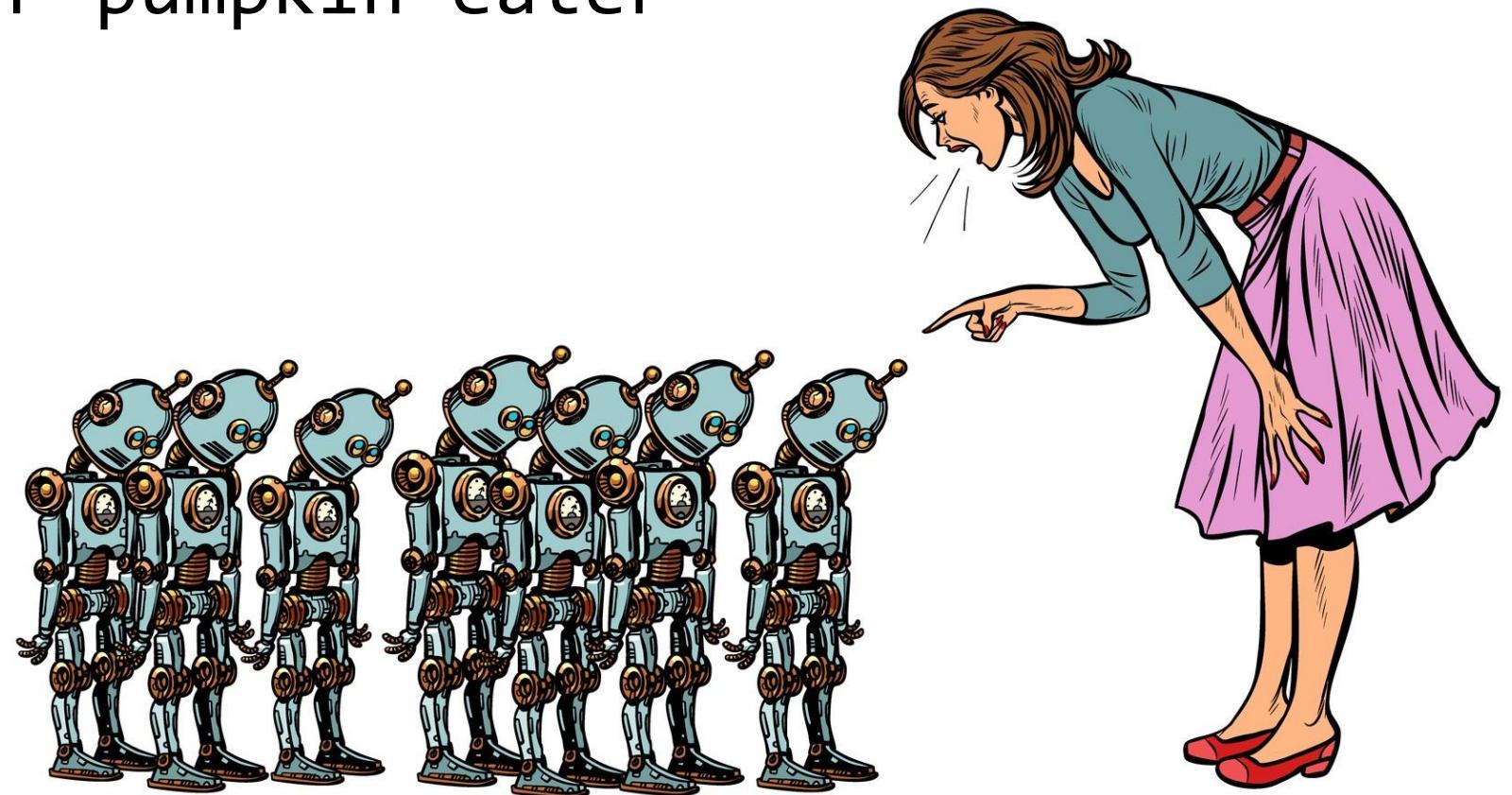
- › Collect network information
- › Run fuzzy string metrics on combos of users and hosts
- › Select next action based on string metric score
 - › Better than random
 - › Better than an if statement

Demo

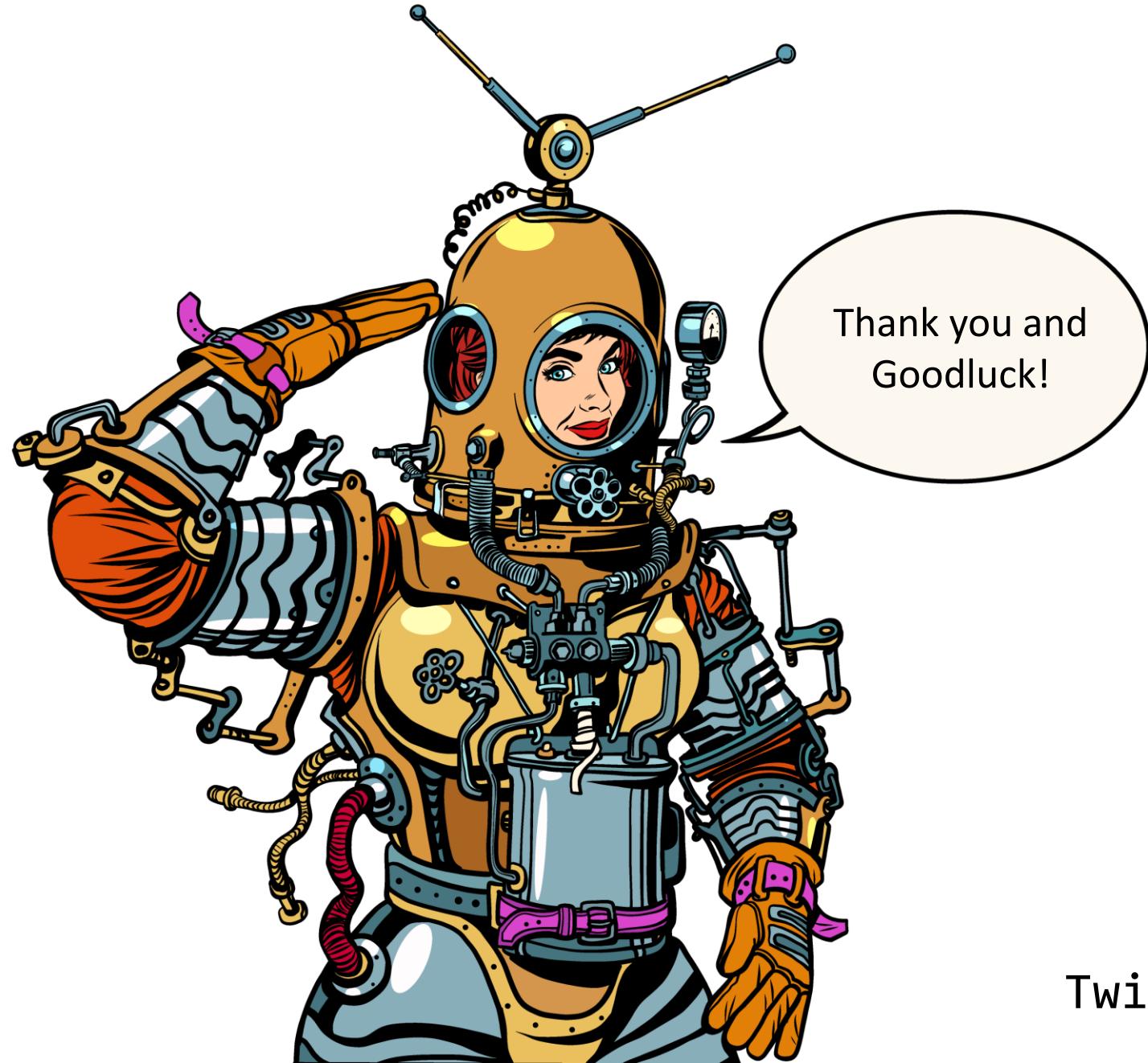


Challenges

- Networks, unknown but discrete
- Cheater cheater pumpkin eater



Big Thank you to Nancy Fulda of BYU



Twitter: @moo_hax