VECTRA®

Security that thinks.®

Data Science in Cybersecurity

ALLAN OGWANG

How Vectra applies data science for threat detection

Vectra uses AI to detect attackers in real time and enrich threat investigations with a conclusive chain of forensic evidence

Automates threat detection

Always-learning behavioral models detect attackers in real-time to enable quick, decisive response and a logical investigative starting point.

Empowers threat hunters

Launch deeper incident investigations detected by Cognito and other security controls, and hunt retrospectively for covert attackers.

Exposes hidden attackers

Network metadata, logs and cloud events are collected, analyzed and stored to reveal hidden attackers in workloads and user/IoT devices.

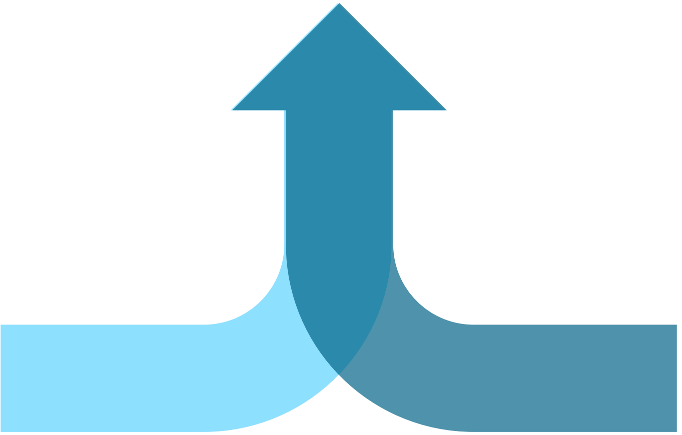
Attacker behaviors: unifying data science and security research

Attacker behavior models

- High-fidelity detection of things attackers must do
- No signatures: find known and unknown

Security Research

- Identify, prioritize, and characterize fundamental attacker behaviors
- Validate models

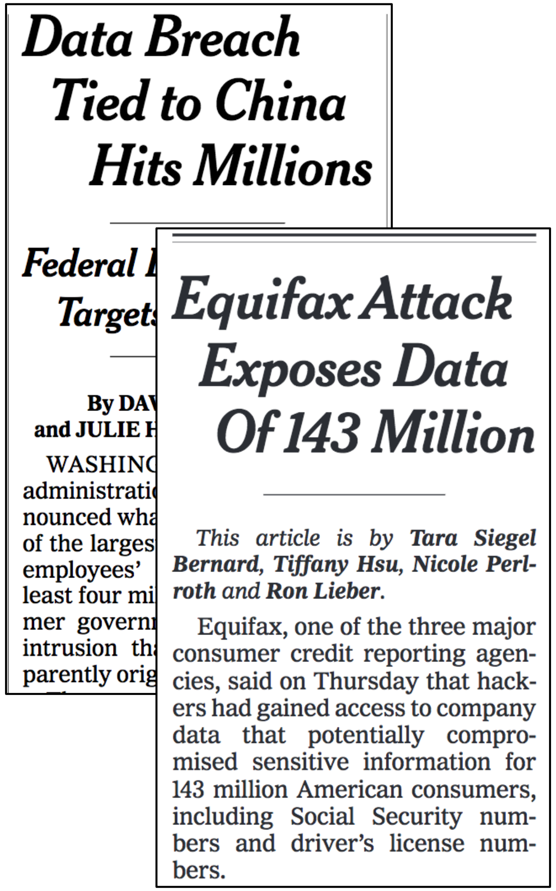


Data Science

- Determine best approach to identify behavior
- Develop and tune models

Who is Vectra AI?

- Vectra AI provides automated threat detection to expose hidden and unknown cyberattackers in a network.
- Apply artificial intelligence to seek out the fundamental threat behaviors that attackers simply can't avoid


Data Breach Tied to China Hits Millions

Federal I
Target

By DAV
and JULIE H

WASHING
administrati
nounced wha
of the larges
employees'
least four mi
mer govern
intrusion th
parently orig

Equifax Attack Exposes Data Of 143 Million

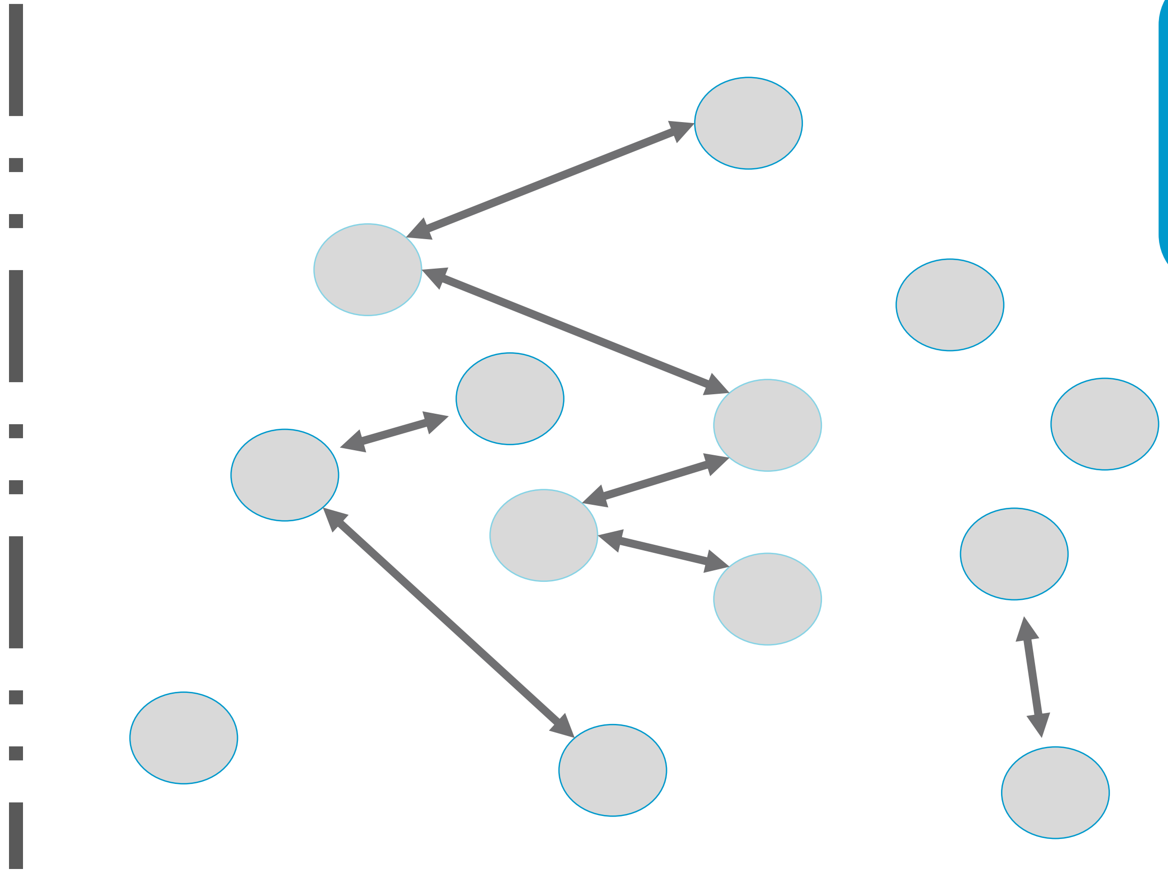
This article is by **Tara Siegel Bernard, Tiffany Hsu, Nicole Perlroth** *and* **Ron Lieber**.

Equifax, one of the three major consumer credit reporting agencies, said on Thursday that hackers had gained access to company data that potentially compromised sensitive information for 143 million American consumers, including Social Security numbers and driver's license numbers.

Cyberthreats in an enterprise: An advanced attack

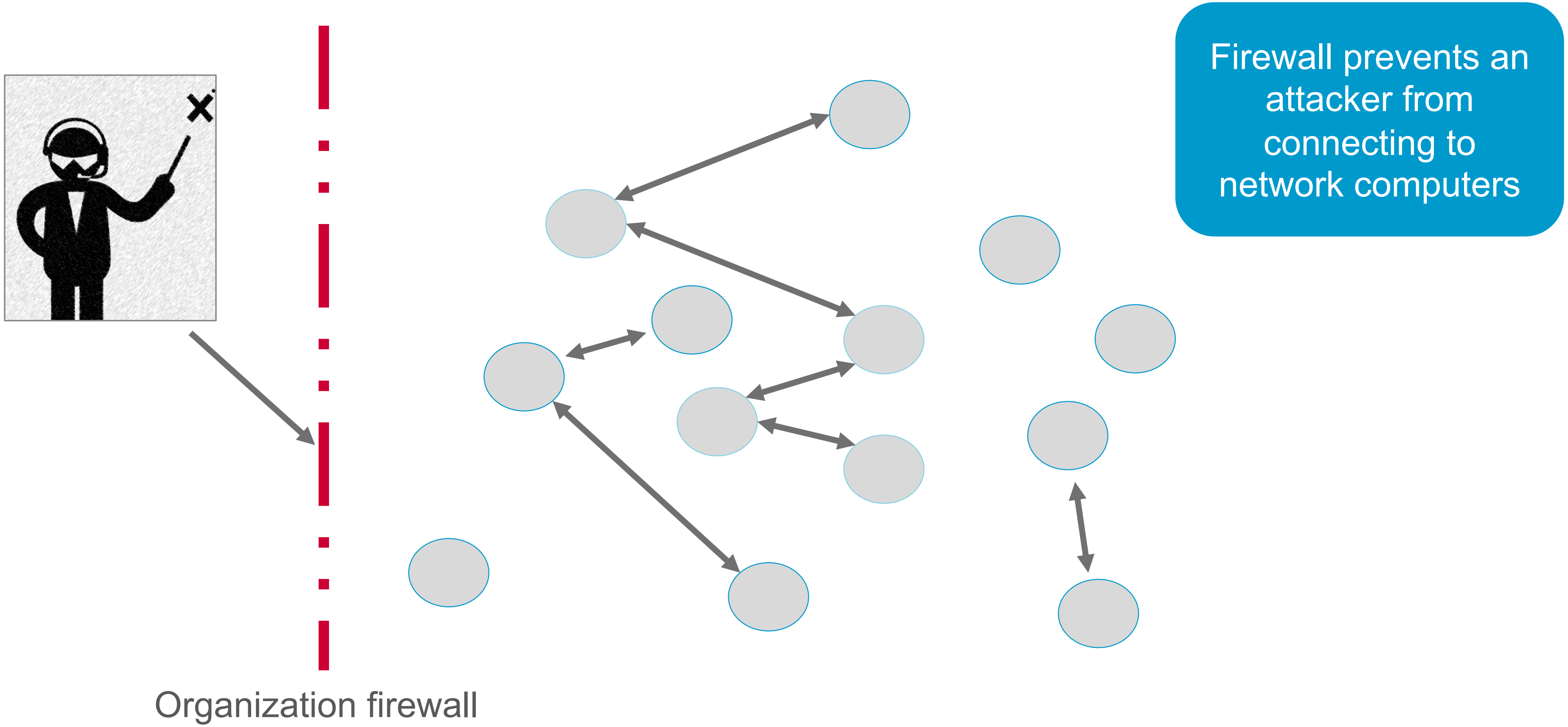
Enterprise networks

Firewall creates a separation between inside and outside of the network

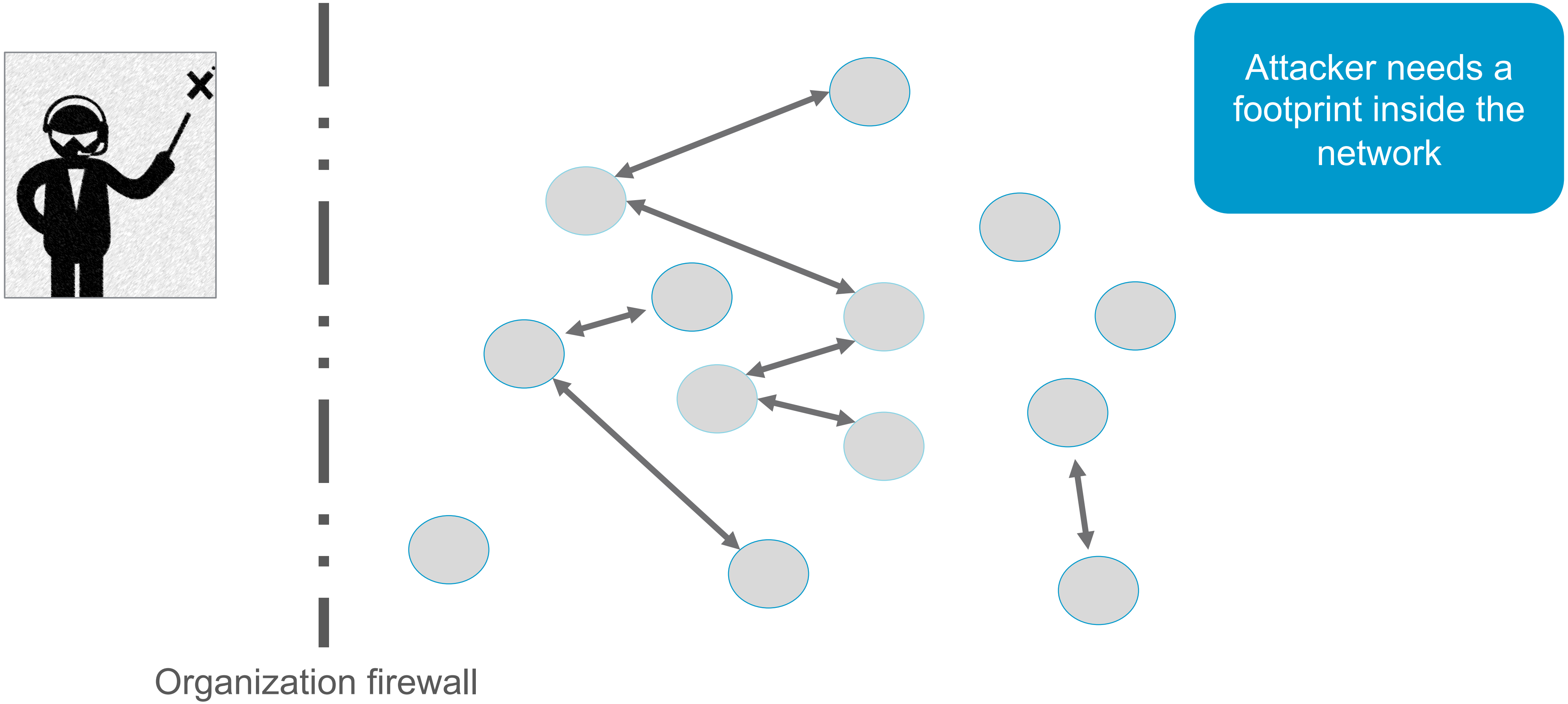


Organization firewall

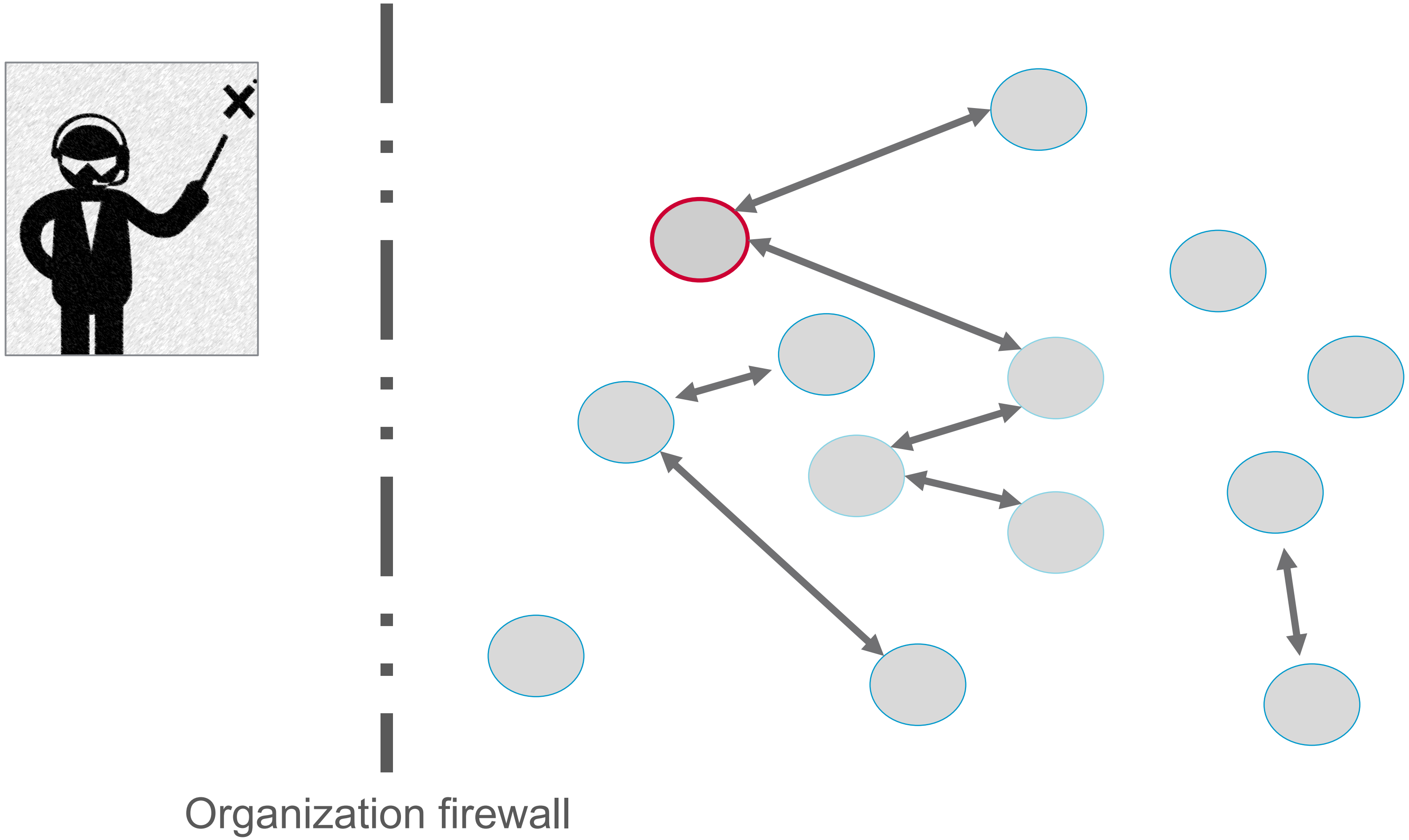
Enterprise networks



Advanced attack

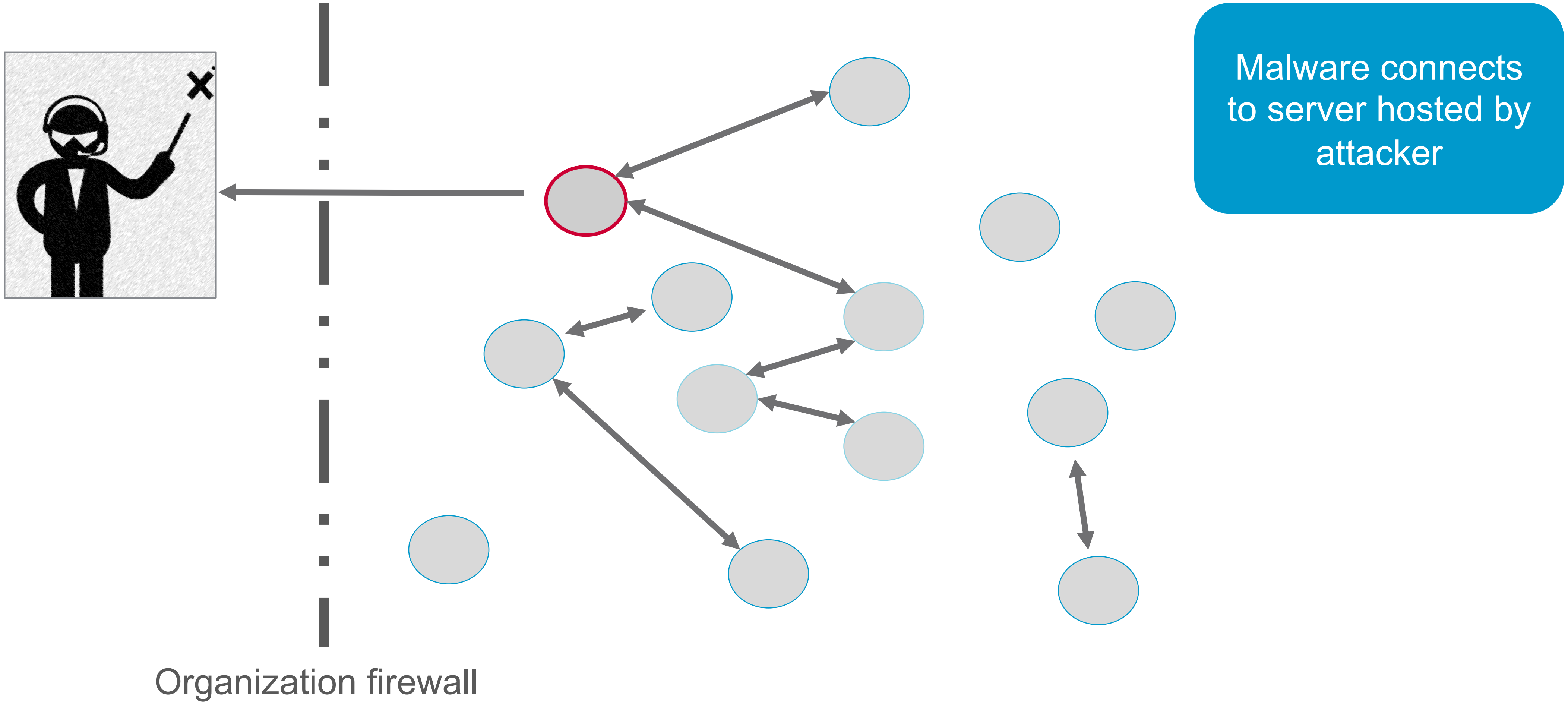


Advanced attack

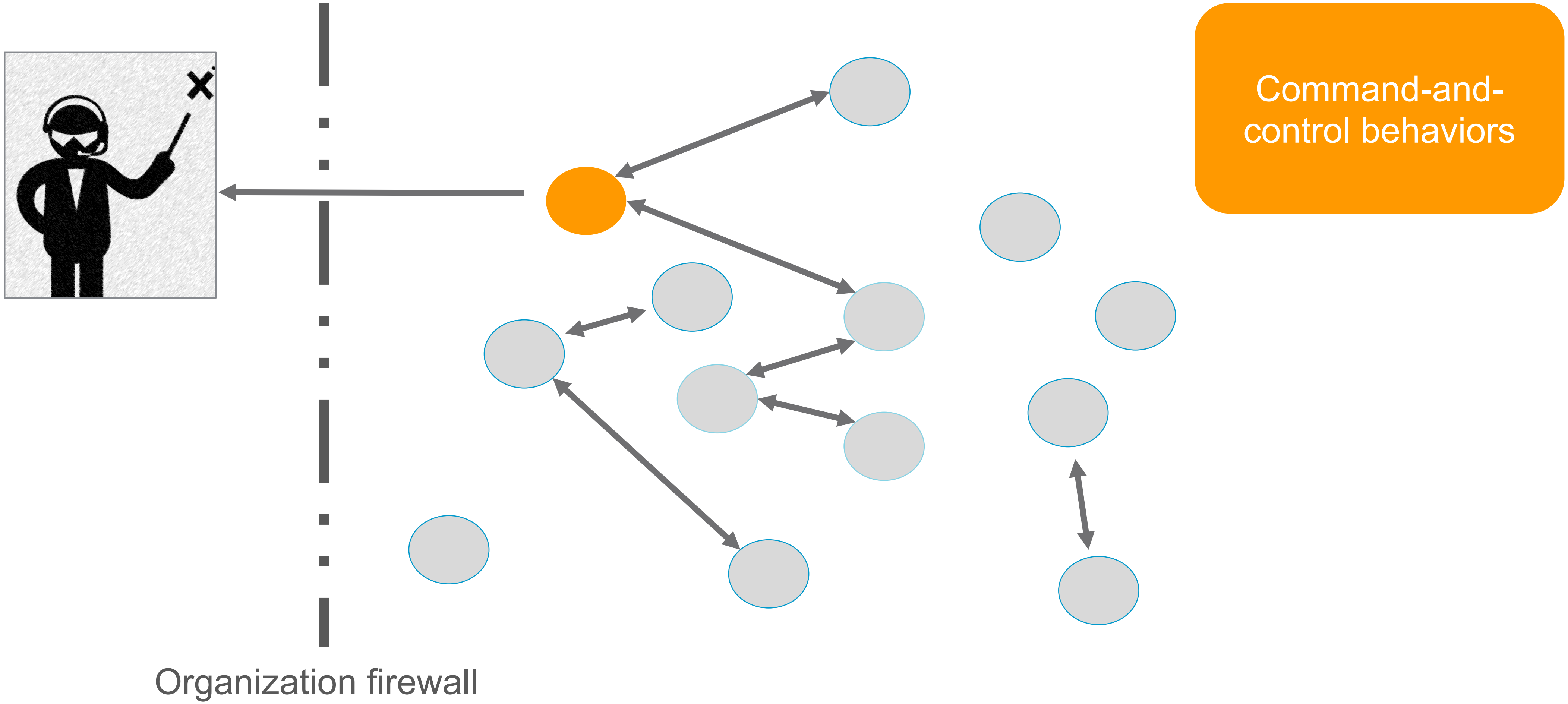


Infect with malware

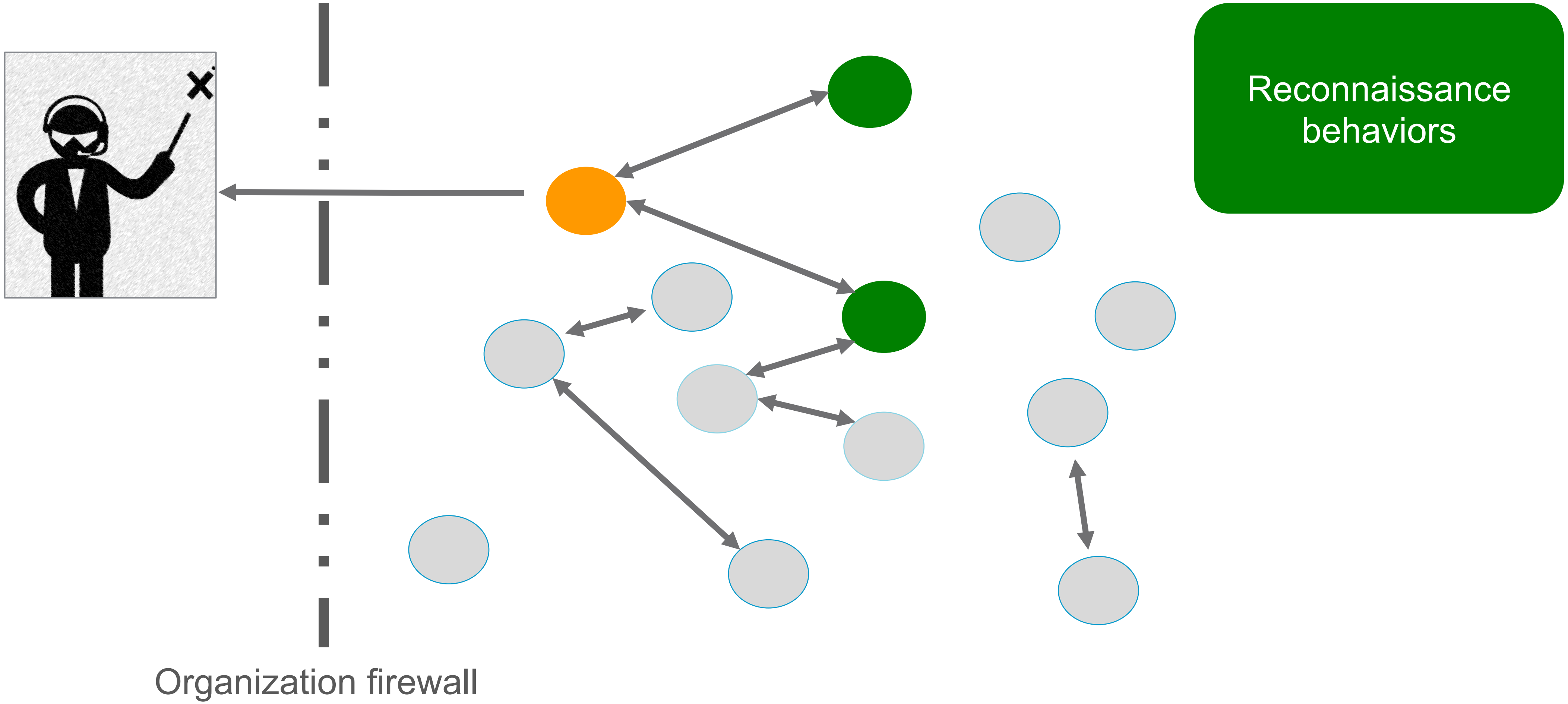
Advanced attack



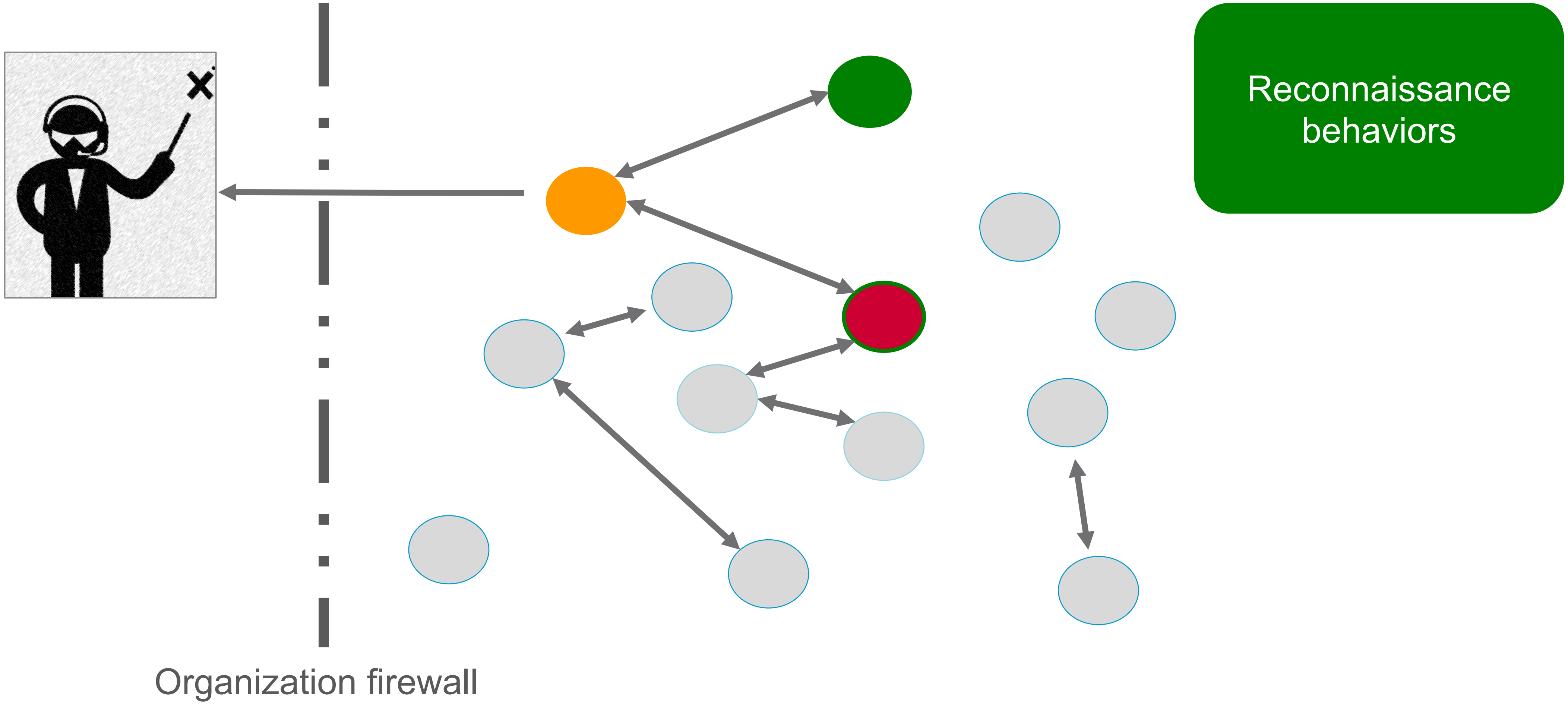
Advanced attack



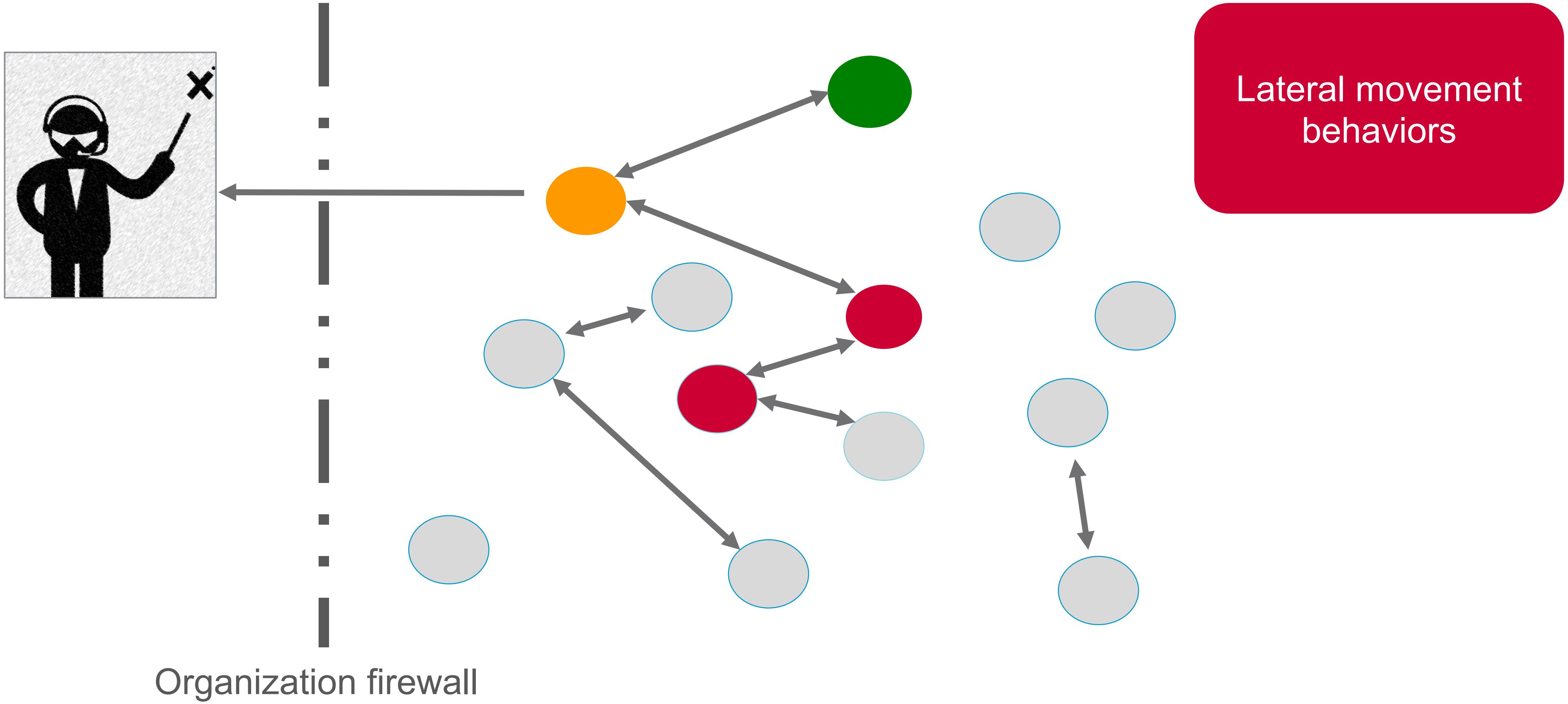
Advanced attack



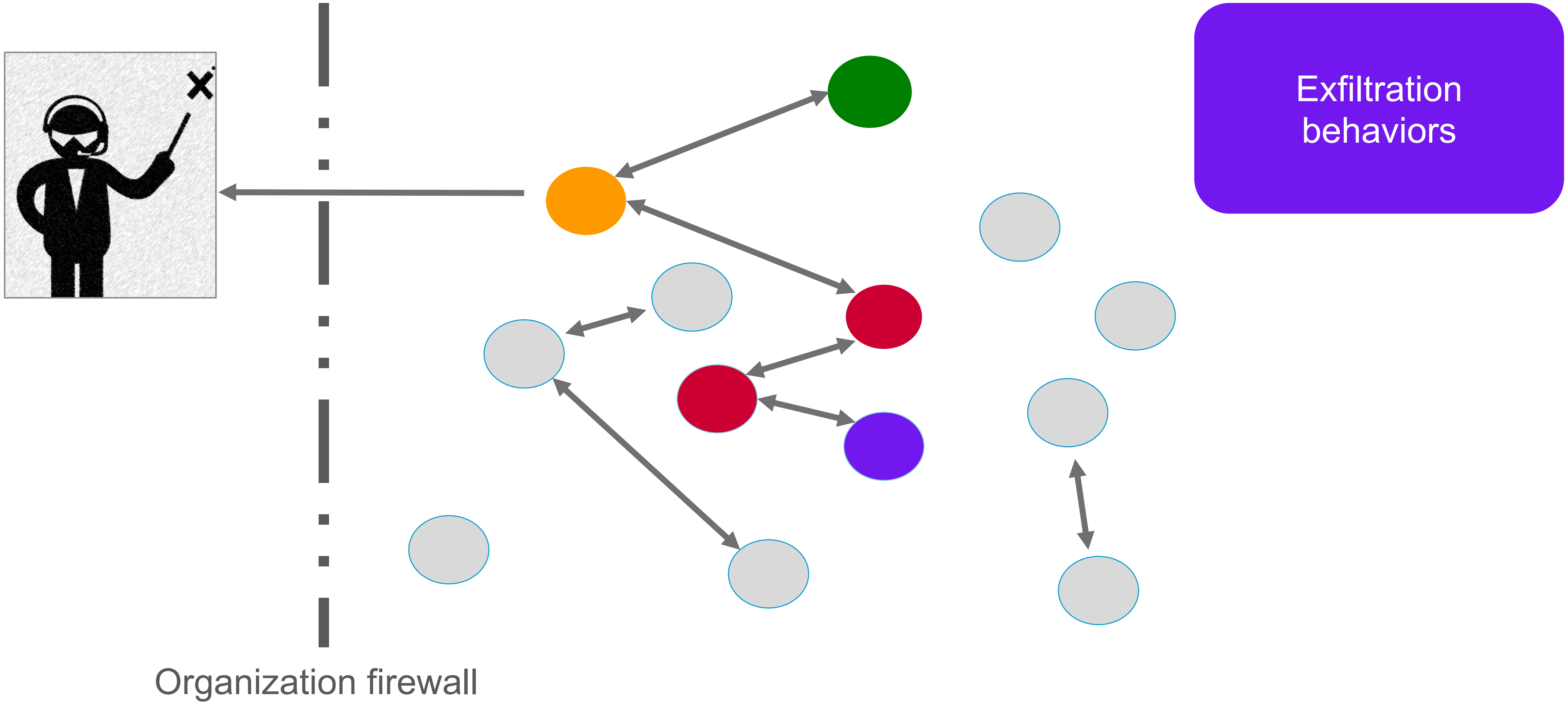
Advanced attack



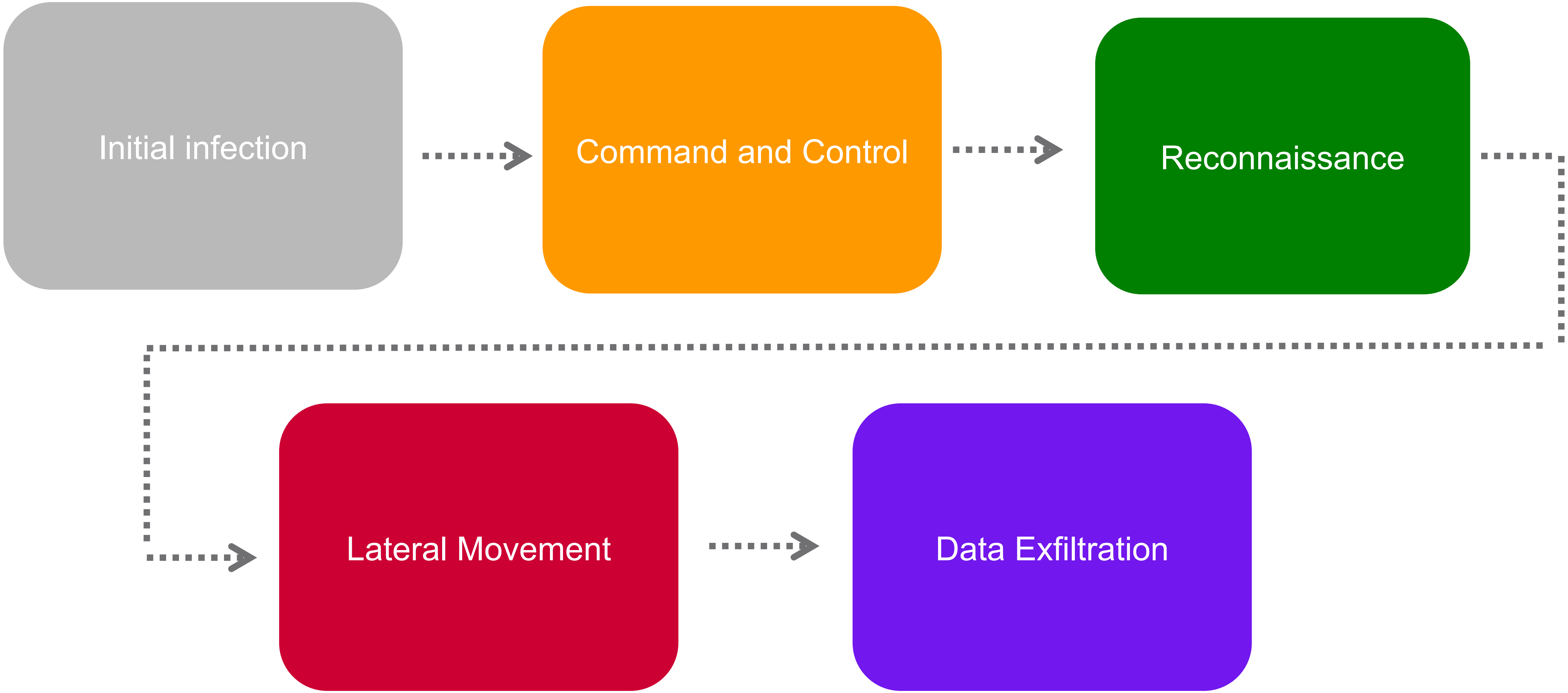
Advanced attack



Advanced attack



Progression of an attack



VECTRA

Attack

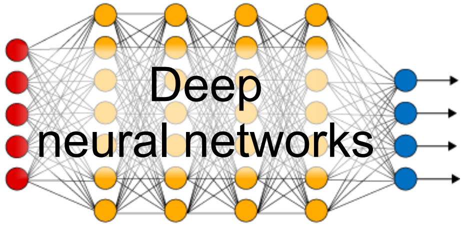
↓

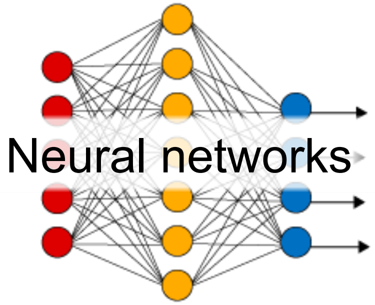
Data

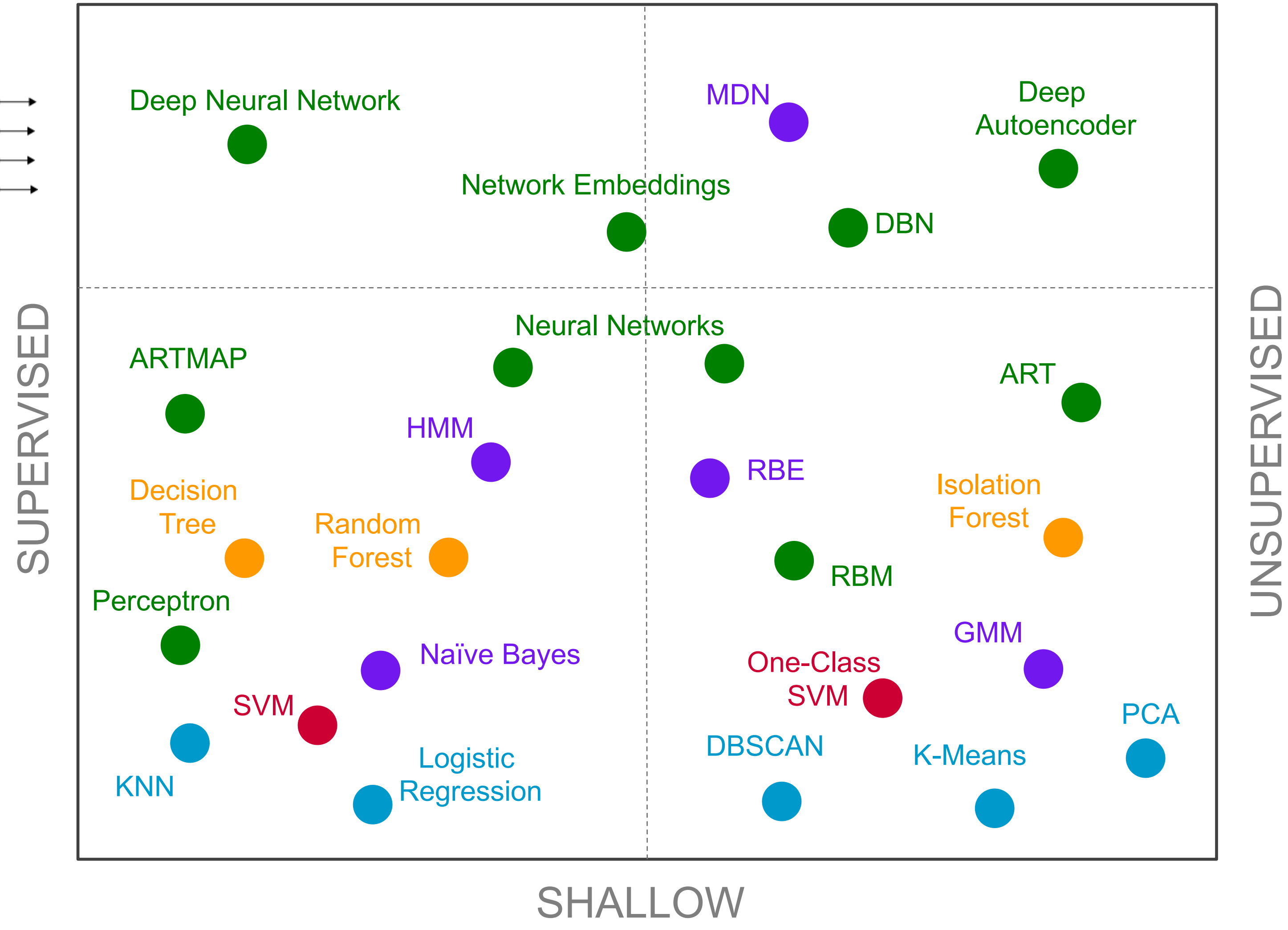
↓

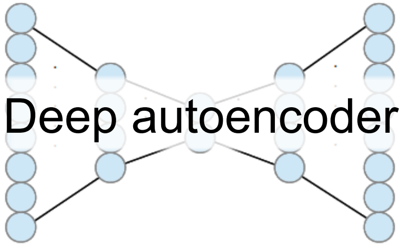
Machine learning

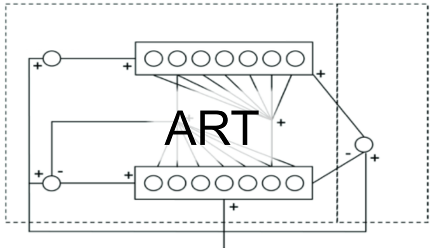
Different types of learning: Supervised vs. unsupervised





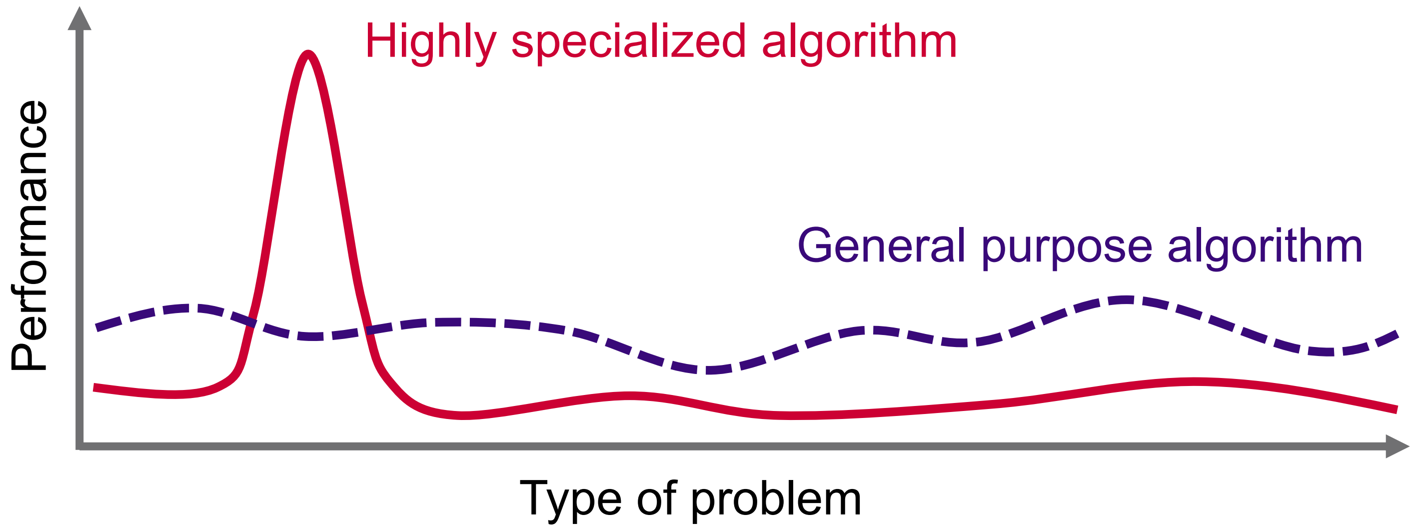




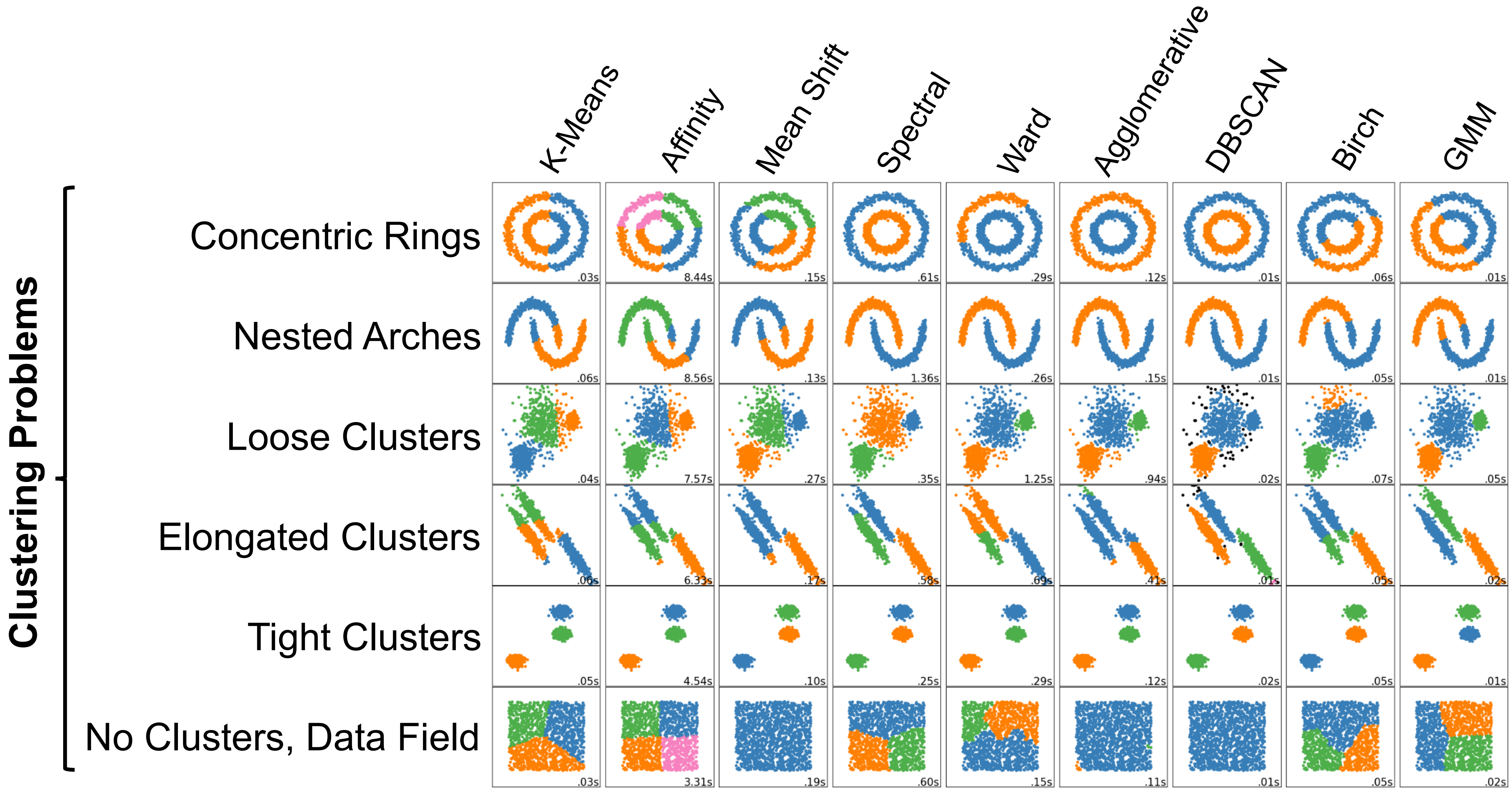


The "no-free-lunch" theorem

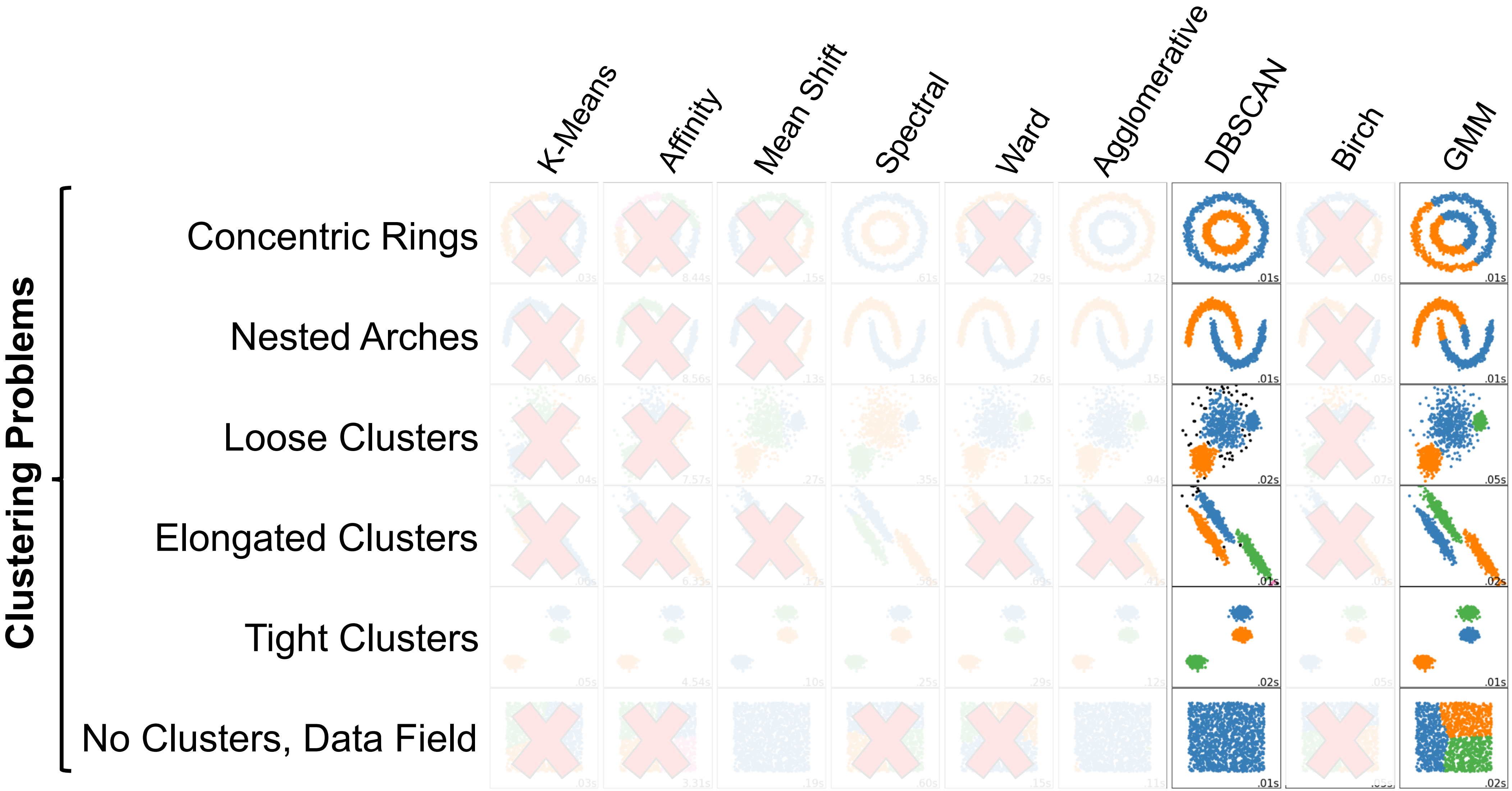
No single algorithm performs best for all problems



Choosing the right algorithm: Know your data

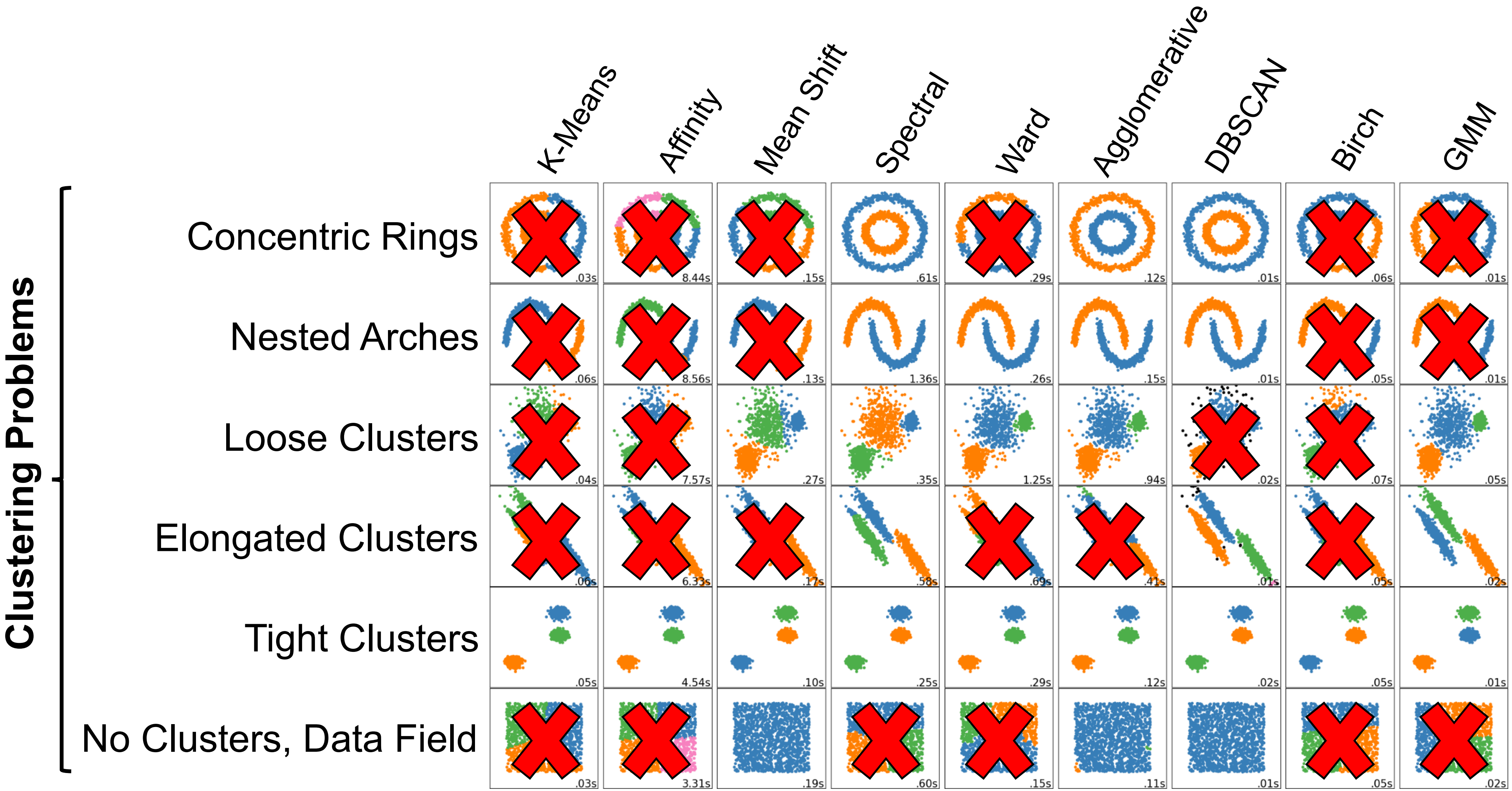


Choosing the right algorithm: Know your data



Choosing the right algorithm: Know your data

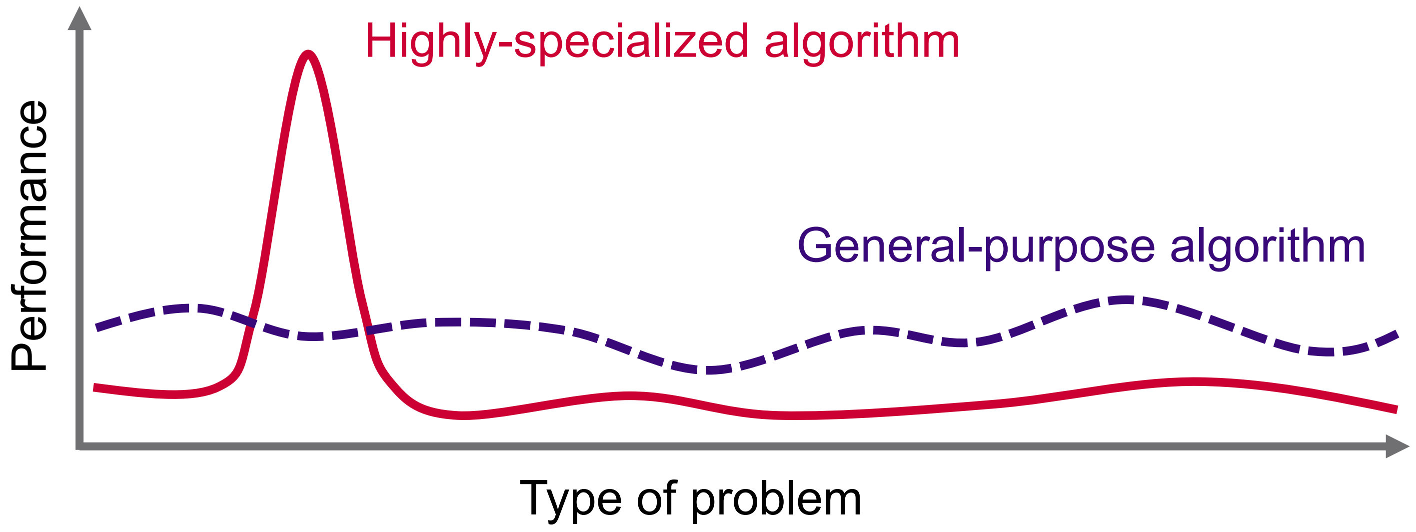
Choosing the right algorithm: Know your data



Choosing the right algorithm

No single algorithm performs best for all problems

Select the right option for your data and performance needs



Outline

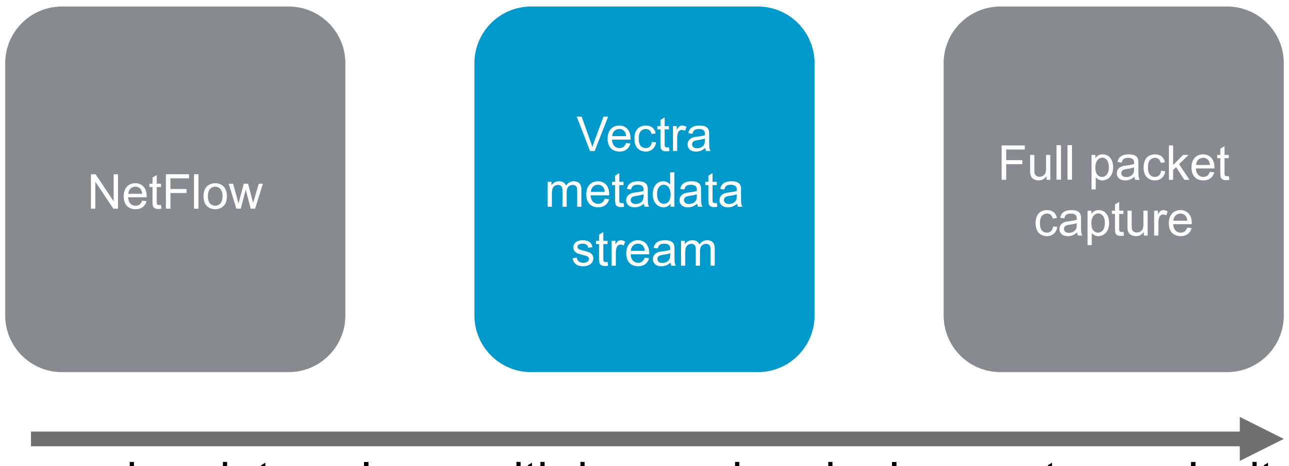
- Metadata used for threat detection
- Approach to detection
  - Detecting Remote Access Trojans (RATs)
    - Signatures
    - Anomaly detection
    - Random forest
    - Deep learning
- Conclusions

Outline

- Metadata used for threat detection
- Vectra's approach to detection
  - Detecting Remote Access Trojans (RATs)
    - Signatures
    - Anomaly detection
    - Random forest
    - Deep learning
- Conclusion

Metadata hits the sweet spot for security applications

- Vectra metadata designed with attacker behavior in mind
- All detection models are based on Vectra metadata
  - Metadata includes bytes, protocols, domains, ips
  - Other advanced models are based off enhanced metadata



Increasing data volume with increasing deployment complexity

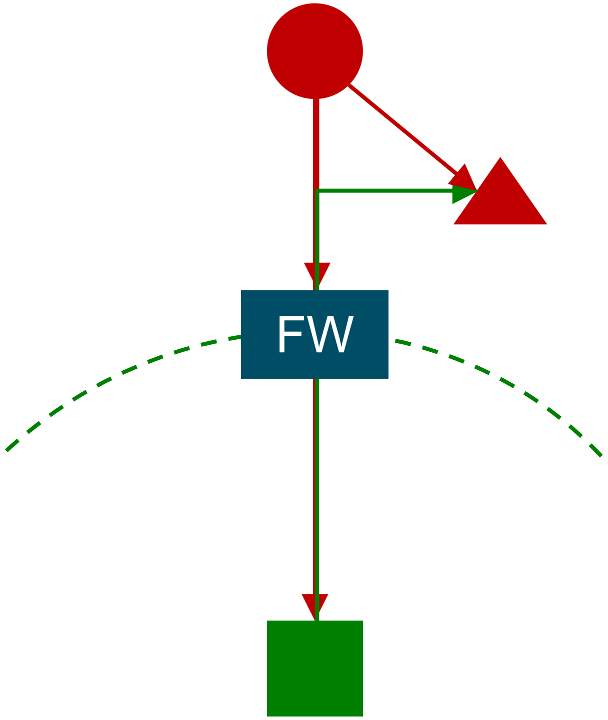
Example of enhanced metadata: Beaconing behavior

- Beaconing behavior is a common sign of a command and control channel
- Whether a host is beaconing must be inferred based on the host behavior
- By applying machine learning to this raw Vectra metadata we can identify beaconing behavior
- HTTP/S tunnel model was developed using this data to help identify command and control channels

Outline

- Metadata used for threat detection
- Approach to threat detection
  - Detecting Remote Access Trojans (RATs)
    - Signatures
    - Anomaly detection
    - Random forest
    - Deep learning
- Conclusion

Remote Access Trojans (aka external remote access)



Attacker wants to establish manual control over asset inside the network

Firewalls block most inbound connection attempts

So compromised internal asset calls out to “meeting point” and attacker takes over

Examples

Blackshades
Poison Ivy
NOPEN (Shadow Brokers)
WebEx
TeamViewer

Network Signatures

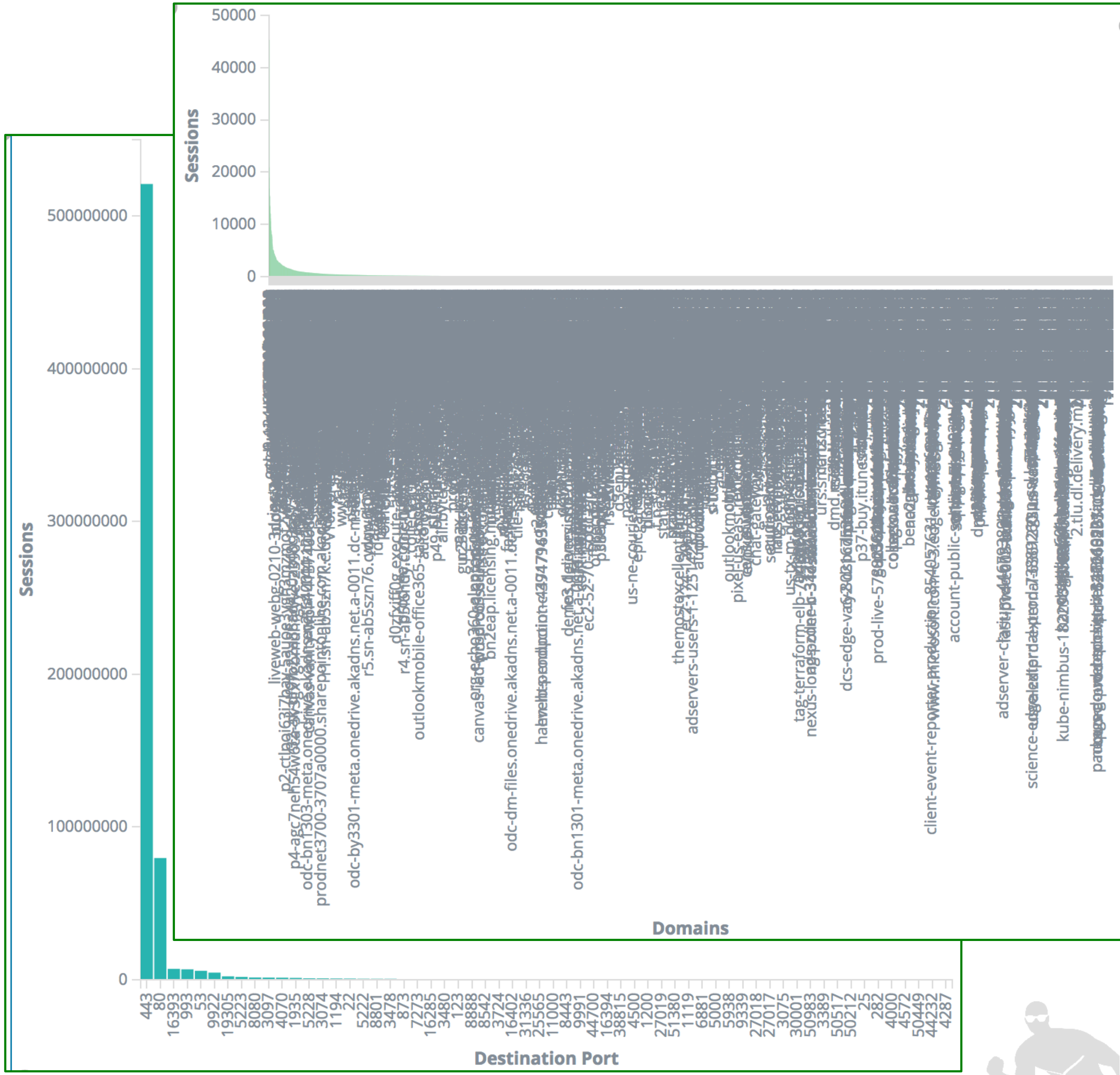
- Based on known patterns flag known RATs
- Network
  - URLS, User Agents, Payloads, Domains, IP Addresses, etc

```
trojan.rules:alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN DarkComet-RAT server join acknowledgement"; flow:to_server,established; dsize:12; content:"|39 34 41 35 41 44 30 41 45 46 36 39|"; flowbits:isset,ET.DarkCometJoin; reference:url,www.darkcometrat.com; reference:url,anubis.iseclab.org/?action=result&task_id=1a7326f61fef1ecb4ed4fbf3de3f3b8cb&format=txt; classtype:trojan-activity; sid:2013284; rev:3; metadata:created_at 2011_07_18, updated_at 2011_07_18;)
```

- Great for known threats
  - Easily bypassed with changes to the malware
  - Lags behind new changes in malware

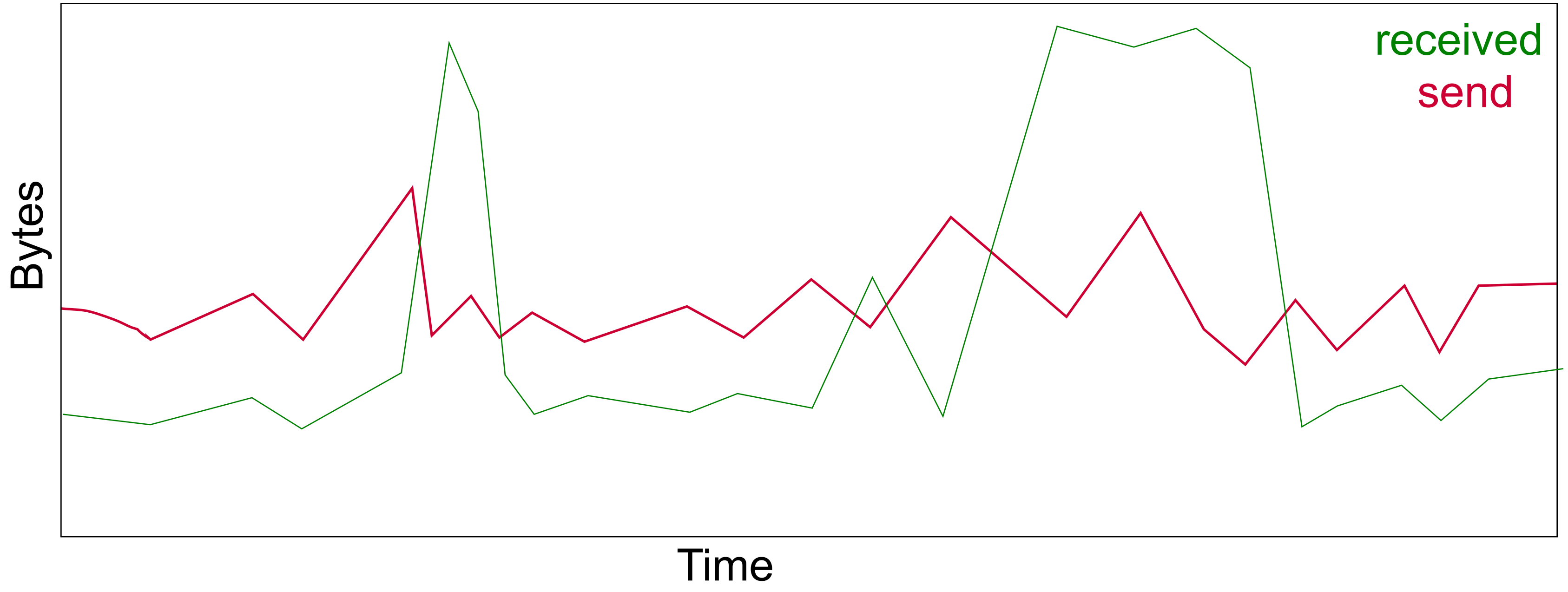
Anomaly detection

- Unsupervised
  - Assume a RAT
    - Uncommonly used port
    - Uncommon destination
    - Uncommon hour
- Everything is “uncommon”
- New ports everyday
- New domains everyday
- Time is not a great signal
- Will likely alert you to the event
  - But how do you find true event in this haystack?

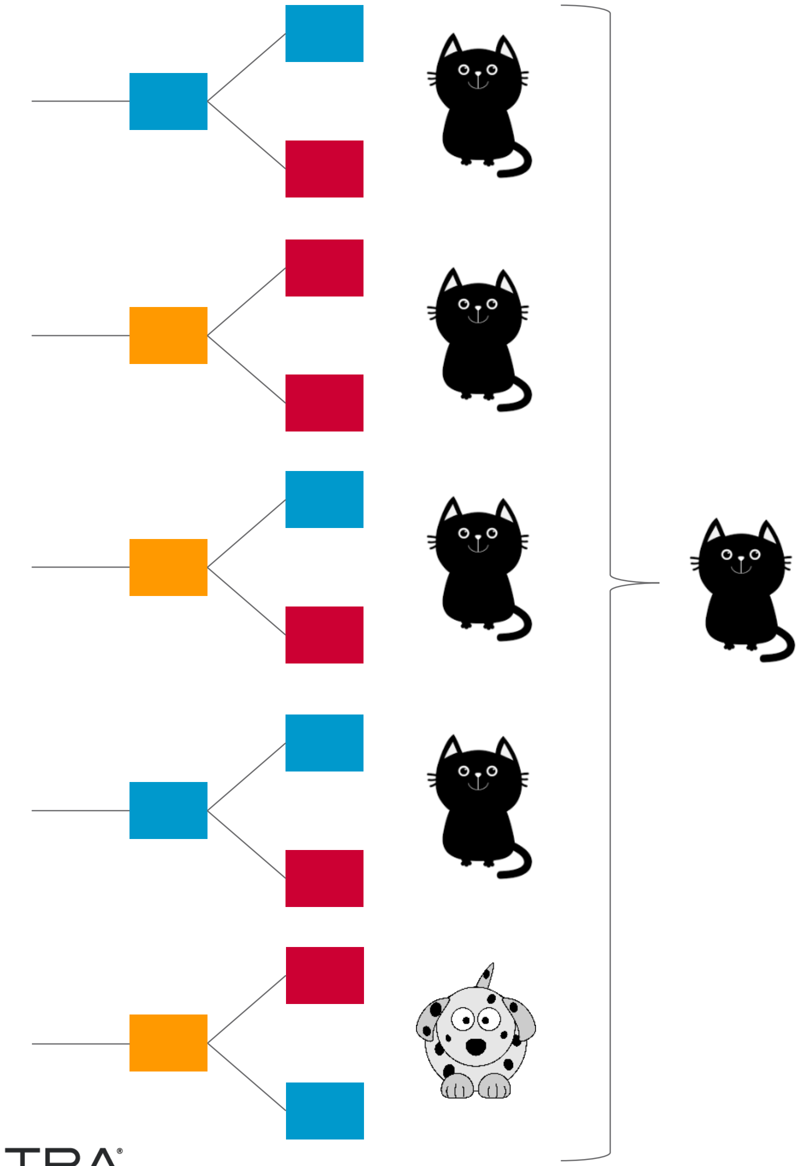


Data is king – How Vectra sees RATs

- A RAT is not static
  - All behavior happens in time
    - Commands are issued
    - Information is received
- Incremental flow between a RAT server and client host



Machine learning first pass – Random forest



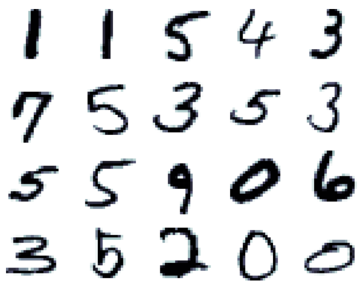
- A random forest is a collection of decision trees
- Not likely a single perfect decision tree model
  - Randomly look at features
  - Randomly look at data
  - Build several models
- Each model votes
  - Every model does not need to be right
  - But more that vote more confidence in decision

Random forest for RATs

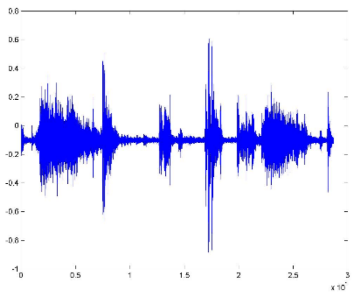
- Featurize the timeseries window – 20+ Features
  - Data and packet client / server ratios
  - Consistency of the client / server data
  - Frequency where the server breaks silence
  - Total session length
  - Entropy of the session
  - etc…
- Observe multiple windows and trigger on convergence

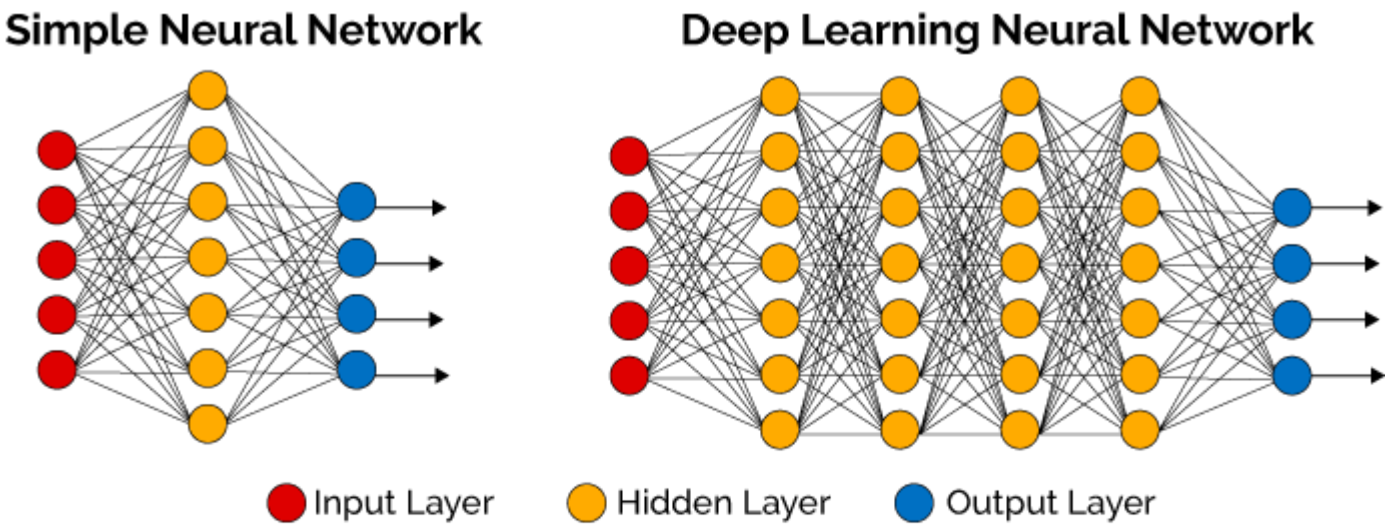
- Model provided value
  - Alerted on large % of known RATs but not all
  - Did not trigger on all known RAT behaviors
- Issues
  - Did not properly represent the temporal nature
    - One sequence impacts the next
  - Human driven features missed behaviors
    - Can guess and test but can never be sure

Deep learning



Digit labels
0,1,2,3,4,5,6,7,8,9



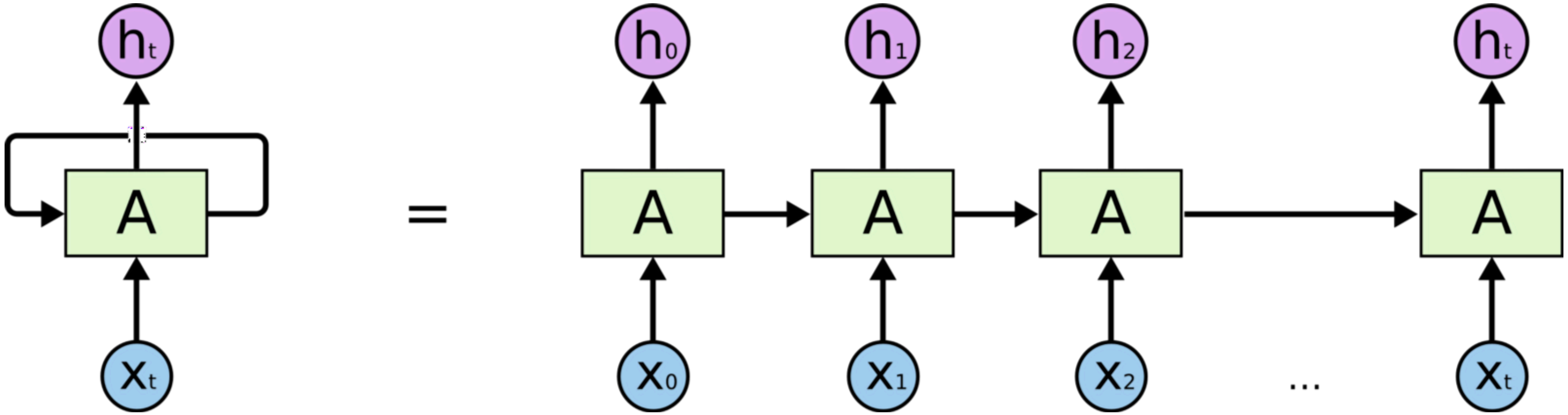


Phonemes
dh, aw, s, ax, n, d, …

Mouse Movements
(right, left, up, down)

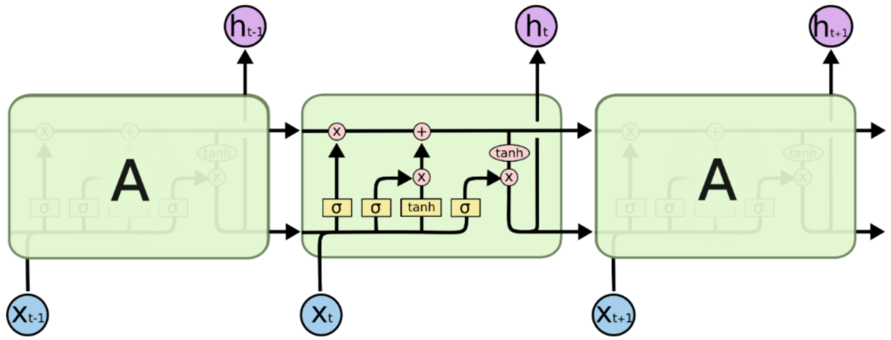
Deep learning:
Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

- RNN
  - Similar to feedforward NN
  - Recurrent connections == Memory

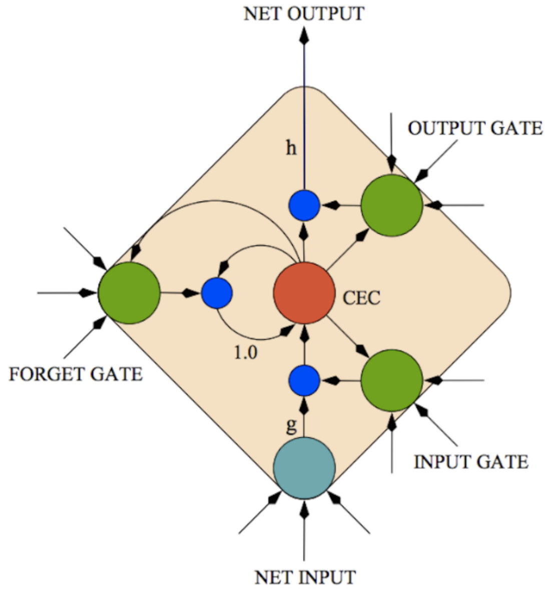


Deep Learning:
Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM)

- LSTM
  - Similar to RNNs
  - Replace simple neurons with LSTM blocks
  - Prevents “vanishing gradient” problem
  - Capable of learning long-range temporal dependencies



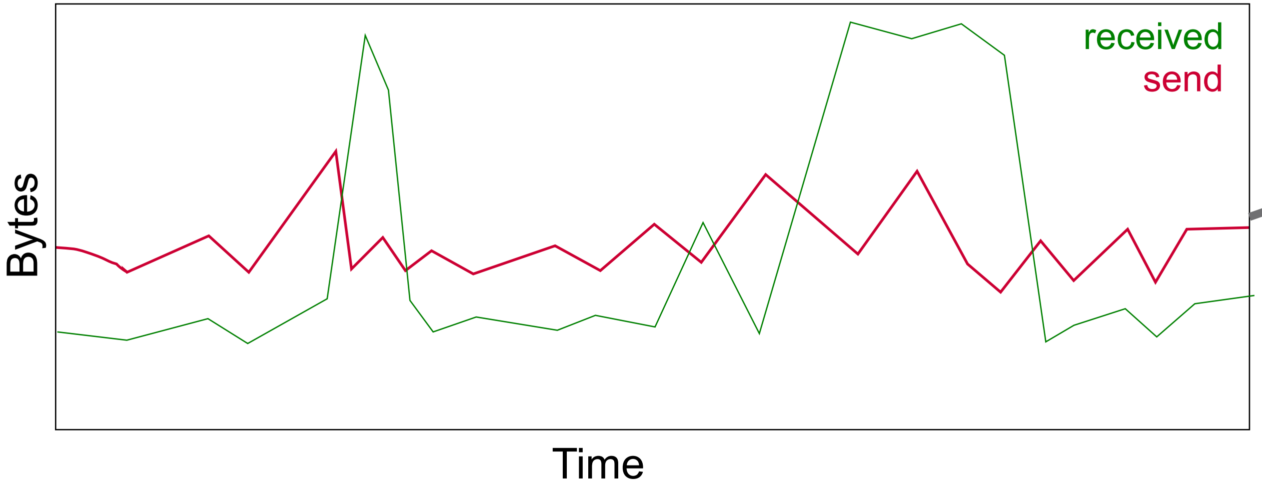
The repeating module in an LSTM contains four interacting layers.

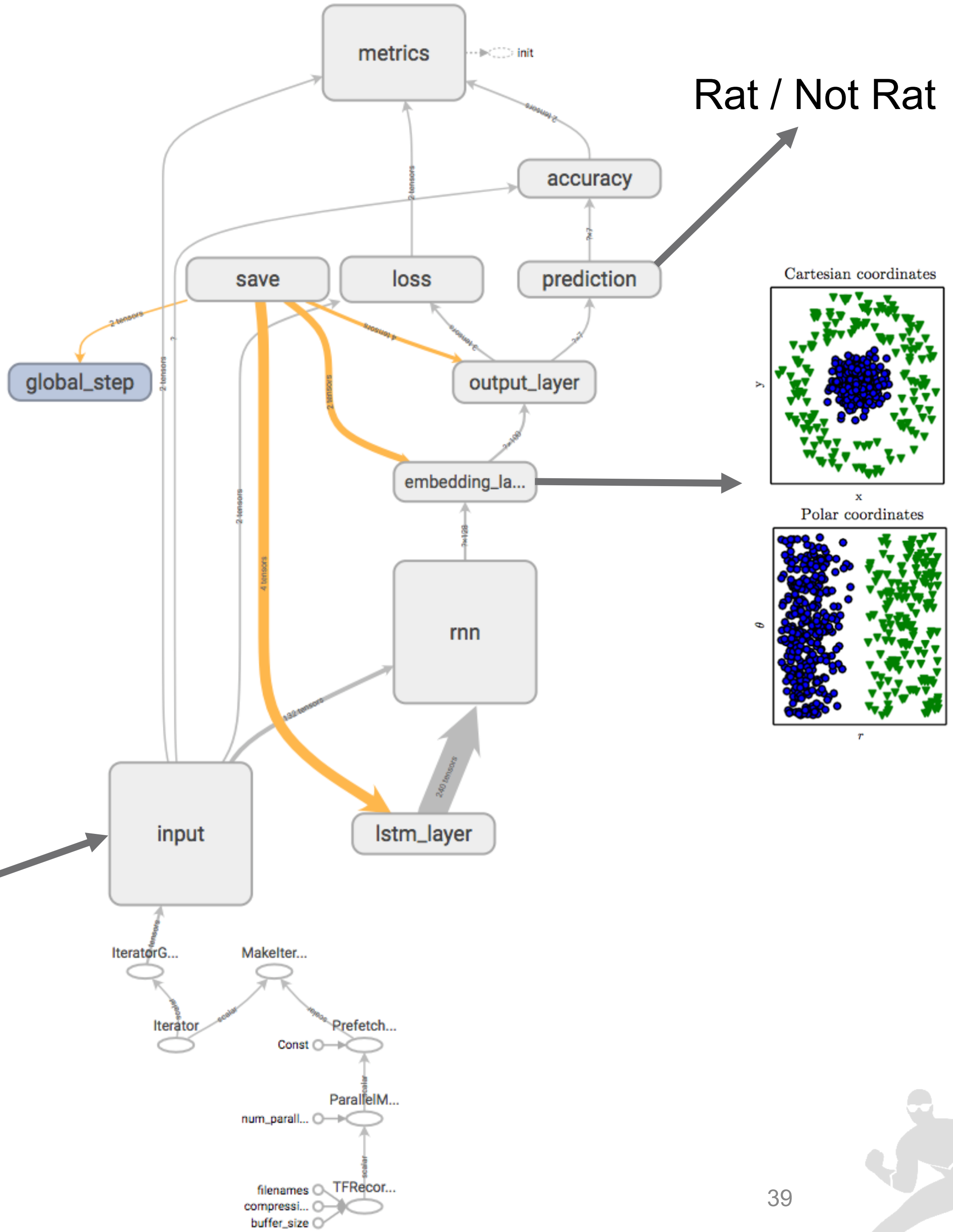


Deep learning:
Model training strategy

Model training

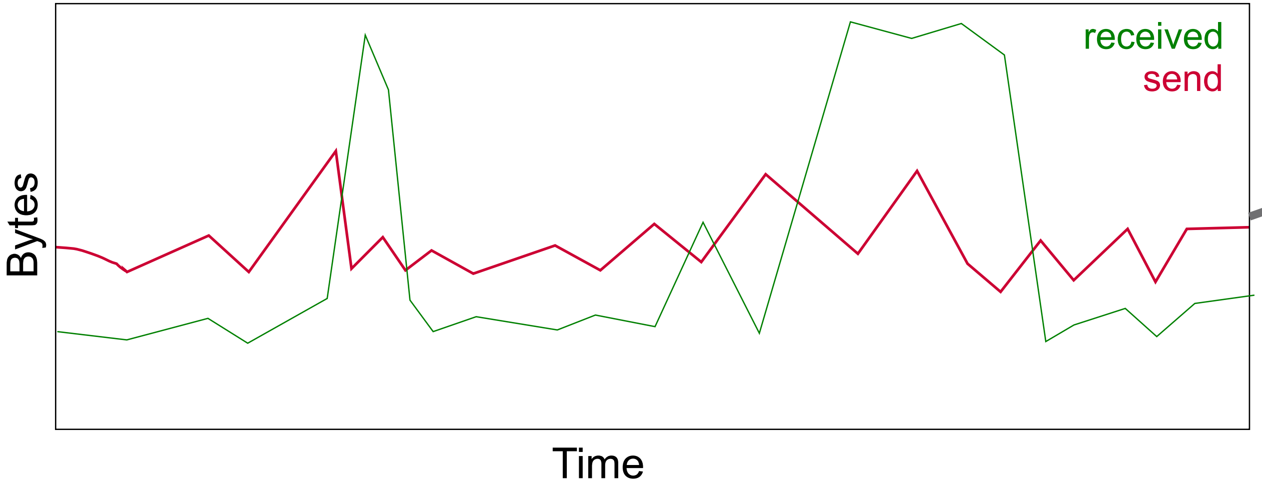
- Framework: TensorFlow
- Model: RNN (LSTM cell)
- Train on AWS w/ NVIDIA v100 GPUs

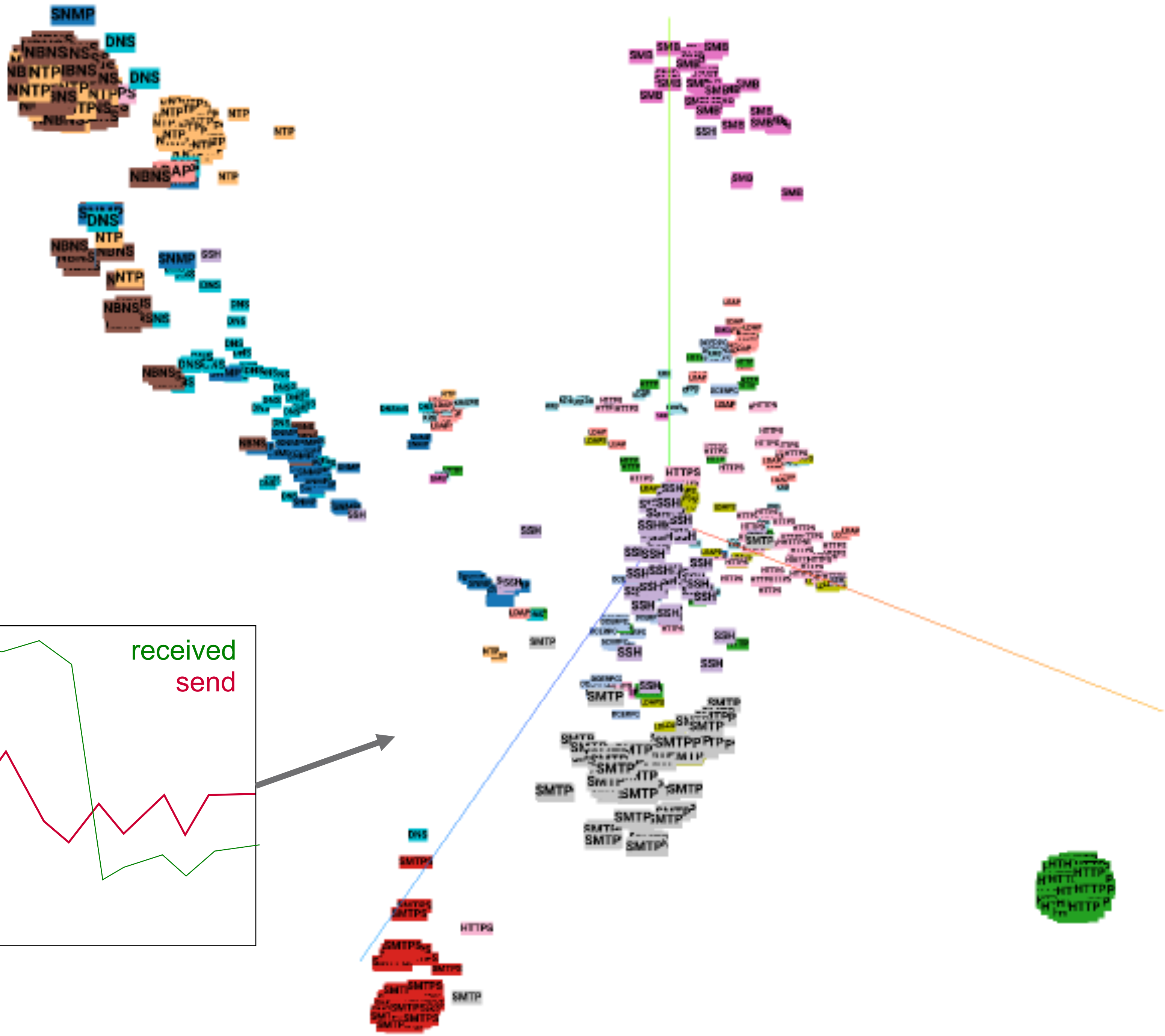




Deep learning:
Learning representations

- Map input time-series to embedded representation
- Classify the embedding as RAT / not RAT
- Observe convergence in classification
- Report behavior





VECTRA

Host: IP-10.1.10.194
IP: 10.1.10.194
Source: Vectra X

External Remote Access

Command & Control

Actions | PCAP | Tag | Note | Assign | Share

Threat 34 / Certainty 10

Summary

Internal Host: IP-10.1.10.194

External Hosts: **54.186.246.98**

Unique Ports: **1**

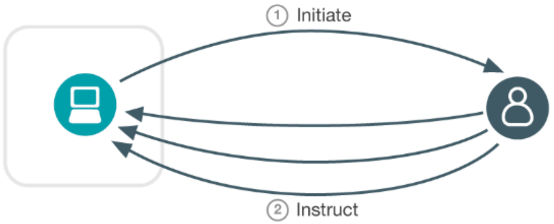
Sessions: **1**

Active Time: **0:16:50**

Bytes Sent: **168.9 KB**

Bytes Received: **77.6 KB**

Infographic



Timeline (Sessions)

Recent Activity

EXTERNAL HOST	PORT	BYTES SENT	BYTES RECEIVED	FIRST SEEN	LAST SEEN
54.186.246.98 ec2-54-186-246-98.us-west-2.compute.amazonaws.com	tcp:22	168.9 KB	77.6 KB	Oct 29th 2018 16:07	Oct 29th 2018 16:24

Outline

- Metadata used for threat detection
- Approach to threat detection
  - Detecting Remote Access Trojans (RATs)
  - Signatures
  - Anomaly detection
  - Random forest
  - Deep learning
- Conclusion

Know your model

- In security, the problems are various and complex; data are sometimes unavailable, sometimes imbalanced

- Many approaches are available, but not all will perform equally well

- No free lunch! Understand the problem and choose the right model
  - Supervised or unsupervised?
  - Classification or regression?
  - Temporal factors are crucial

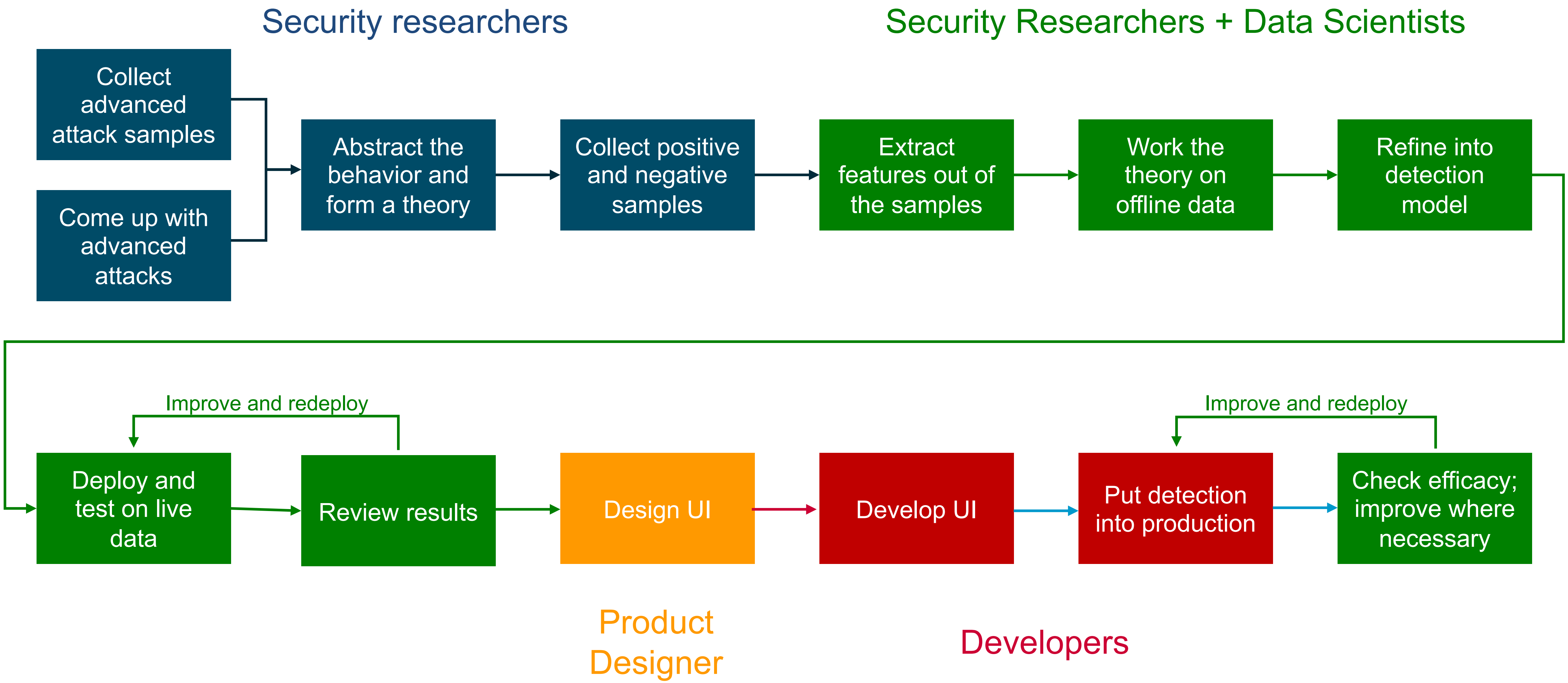
- Data science is not just about math. Attackers can only be detected through conjunction of deep knowledge of machine learning and security

Data science – first as an art, then apply the science

VECTRA®

Security that thinks.®

Detection lifecycle



Model Development Philosophy – *Research to Production*

1. Report an advanced attack behavior
   - Methodology and data sources are irrelevant

2. Provides the relevant context to investigate
   - Necessary information for rapid validation

3. Improvable over time
   - Trackable efficacy

4. Minimal noise and high coverage
   - Meets initial recall and precision requirements