Web Shell Case Study

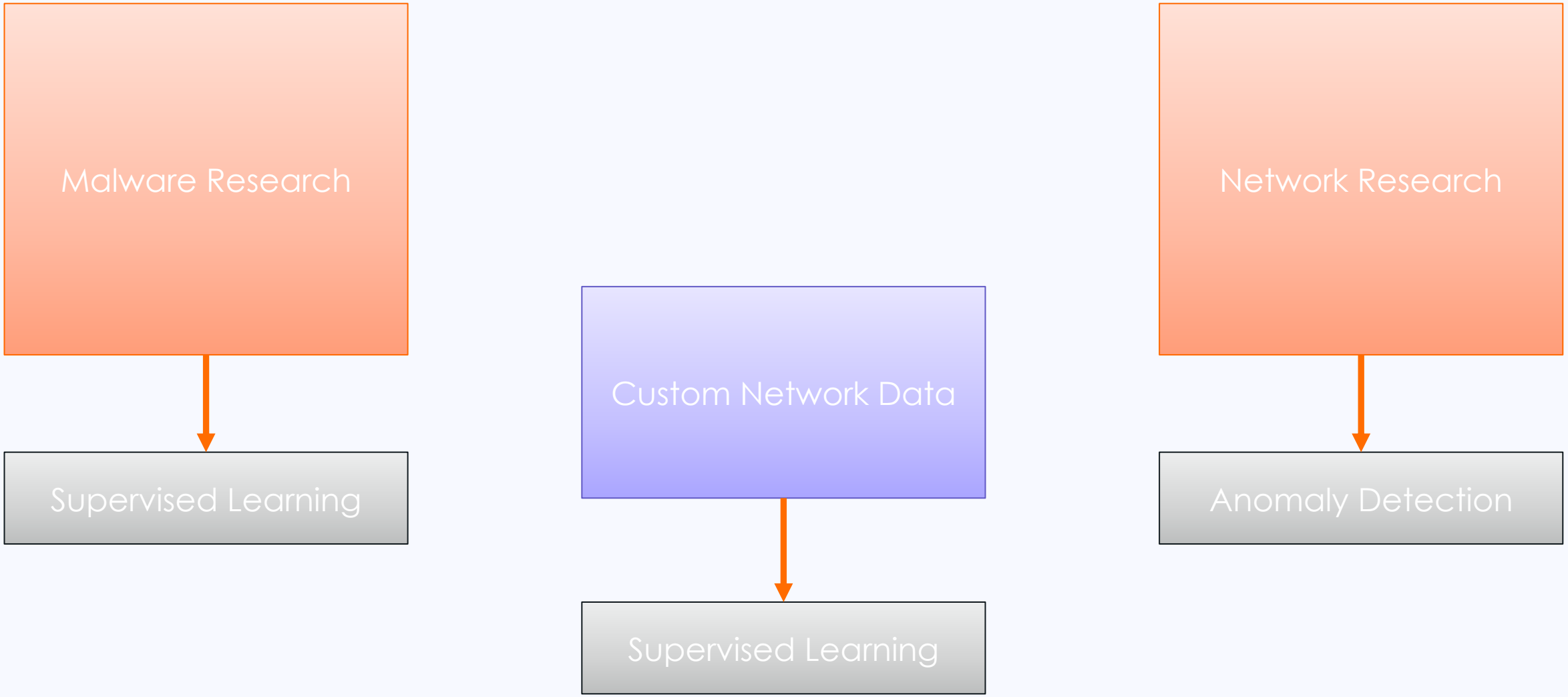
Lindsey Lack
John Conwell

Applied Threat Research
Gigamon Threat Insight

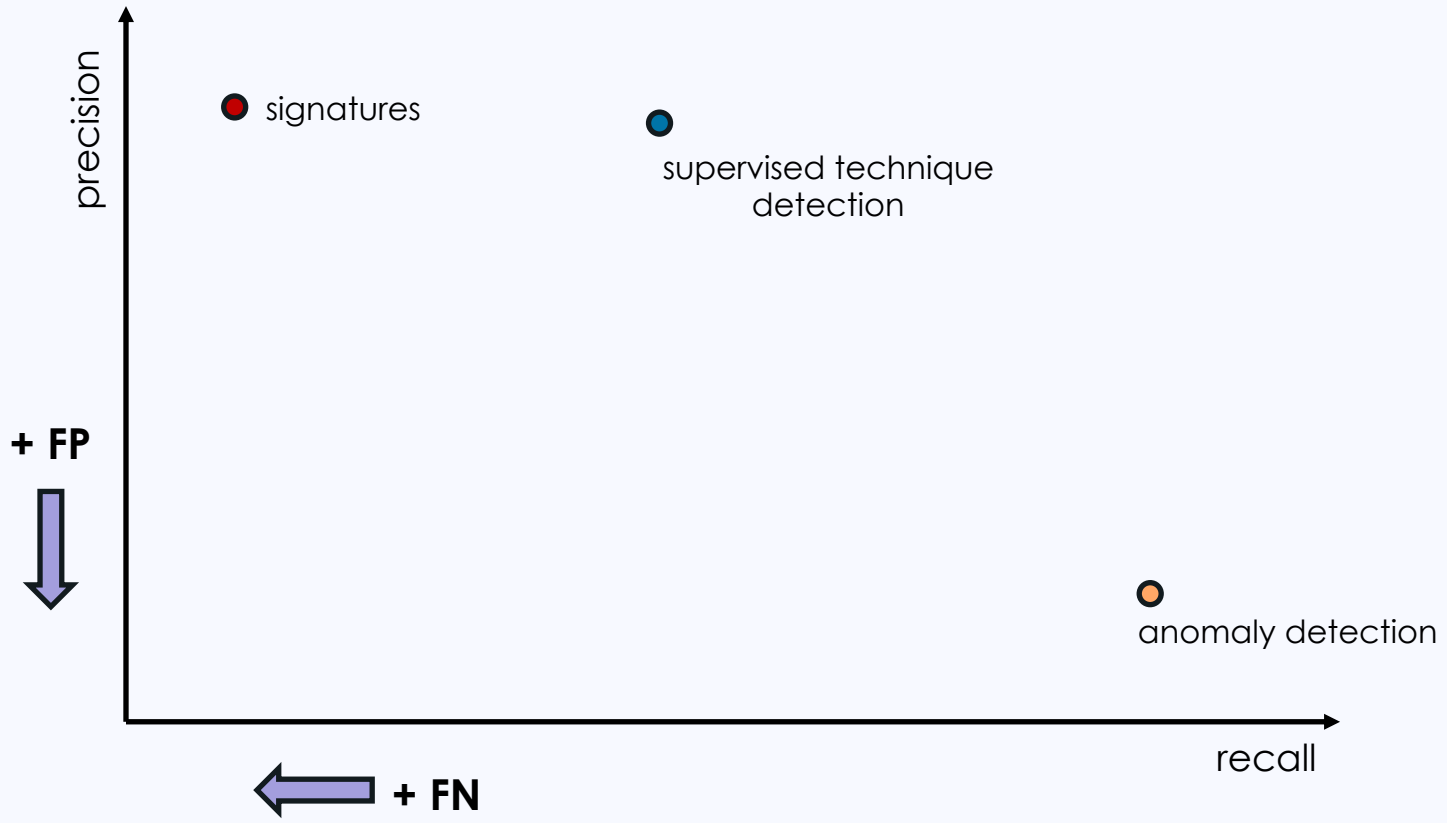
© 2017-2022 Gigamon Inc. All rights reserved.

Gigamon®

- Motivation
- Web Shell 101
- Setup
- Case Study with tips and guidelines


Malware Research
Supervised Learning
Custom Network Data
Supervised Learning
Network Research
Anomaly Detection

Objective


precision
signatures
supervised technique
detection
anomaly detection
+ FP
+ FN
recall

Webshell 101

Setup

Assumptions

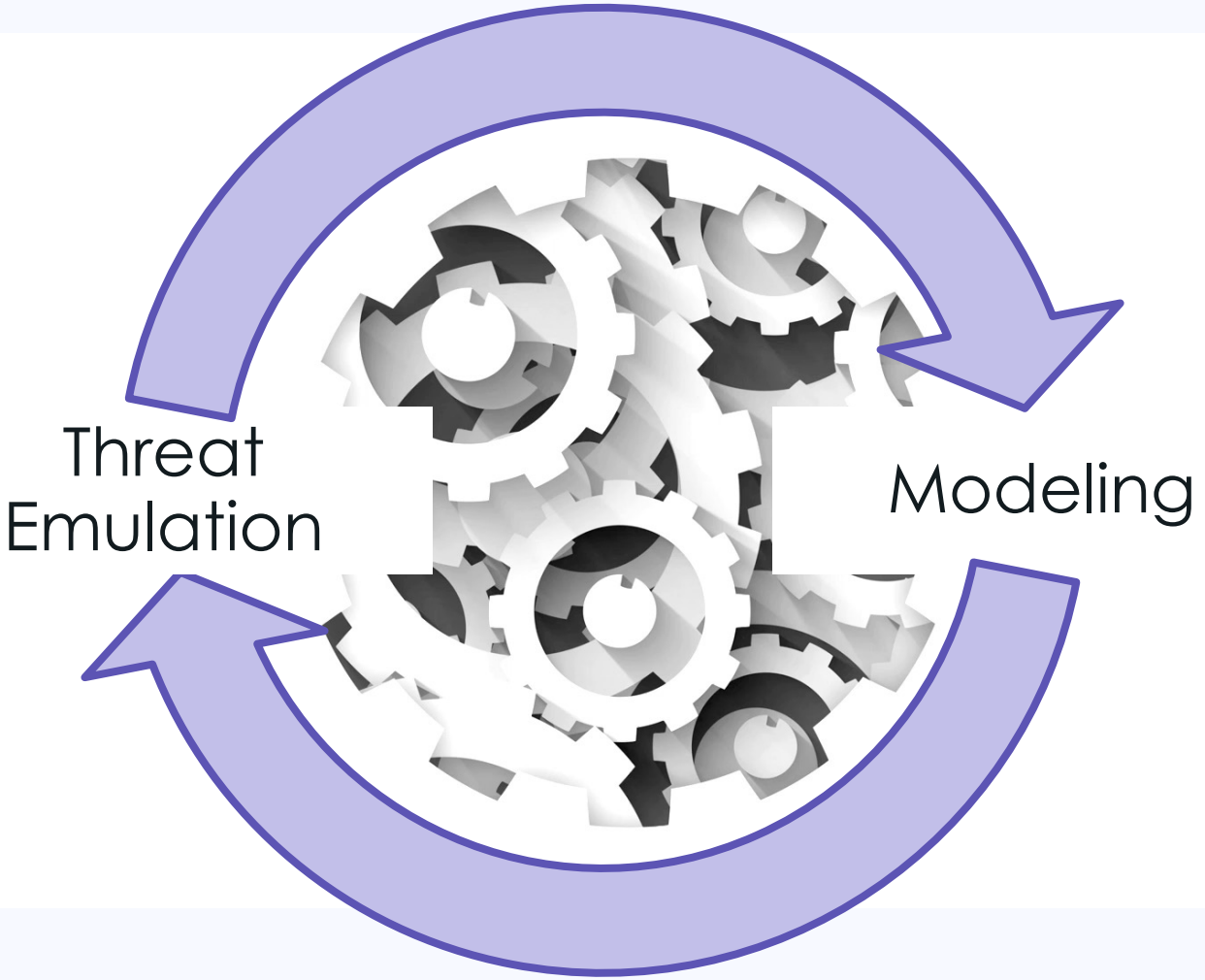
- Enterprise volume network data
  - At least North-South being captured
- Zeek-style metadata being generated and available

Problem Scoping

- Limit to external sources and internal servers
- Limit to HTTP
- Limit to web shell hosted by new file

Threat Emulation

- Started with 8 manufactured web shell episodes
  - Behinder, ASPXSpy, Godzilla, antsword, etc
  - Create PCAPs from lab environment
  - Extract Zeek metadata



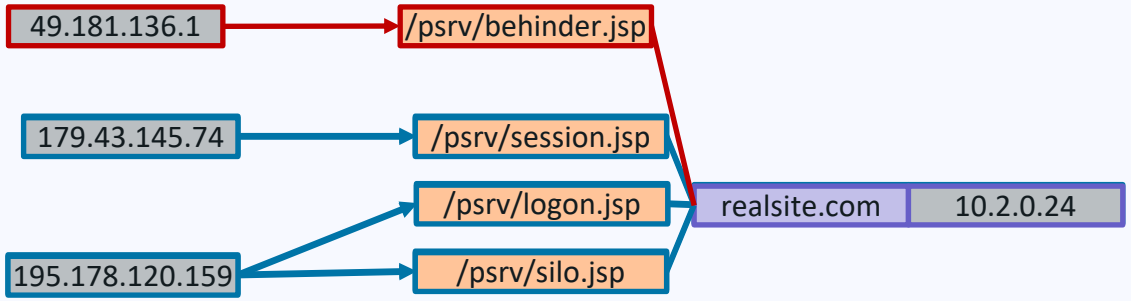
© 2017-2022 Gigamon Inc. All rights reserved.

Hiding the Needle

Effectively weaving sample data into real network traffic

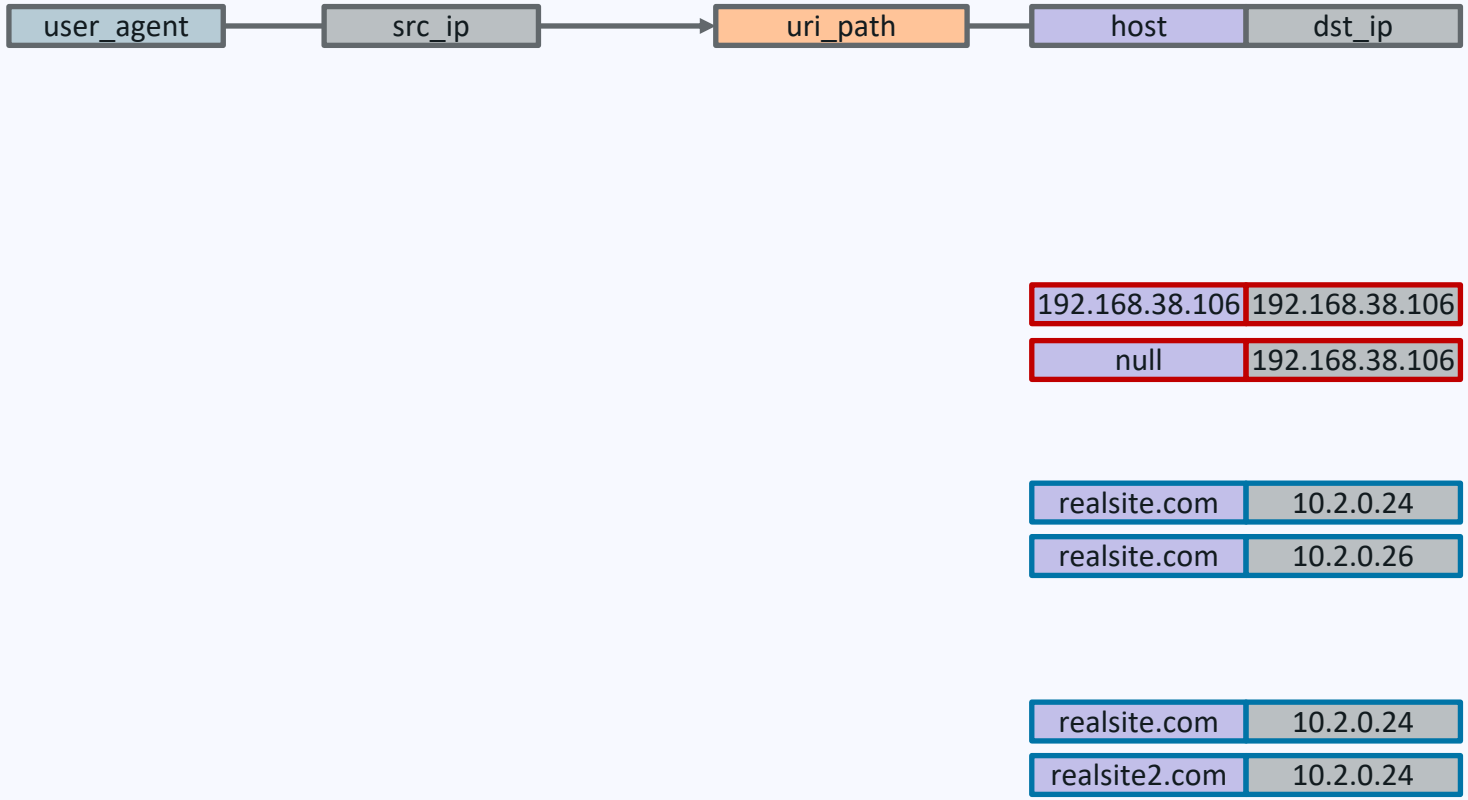
Emulated traffic

Background traffic



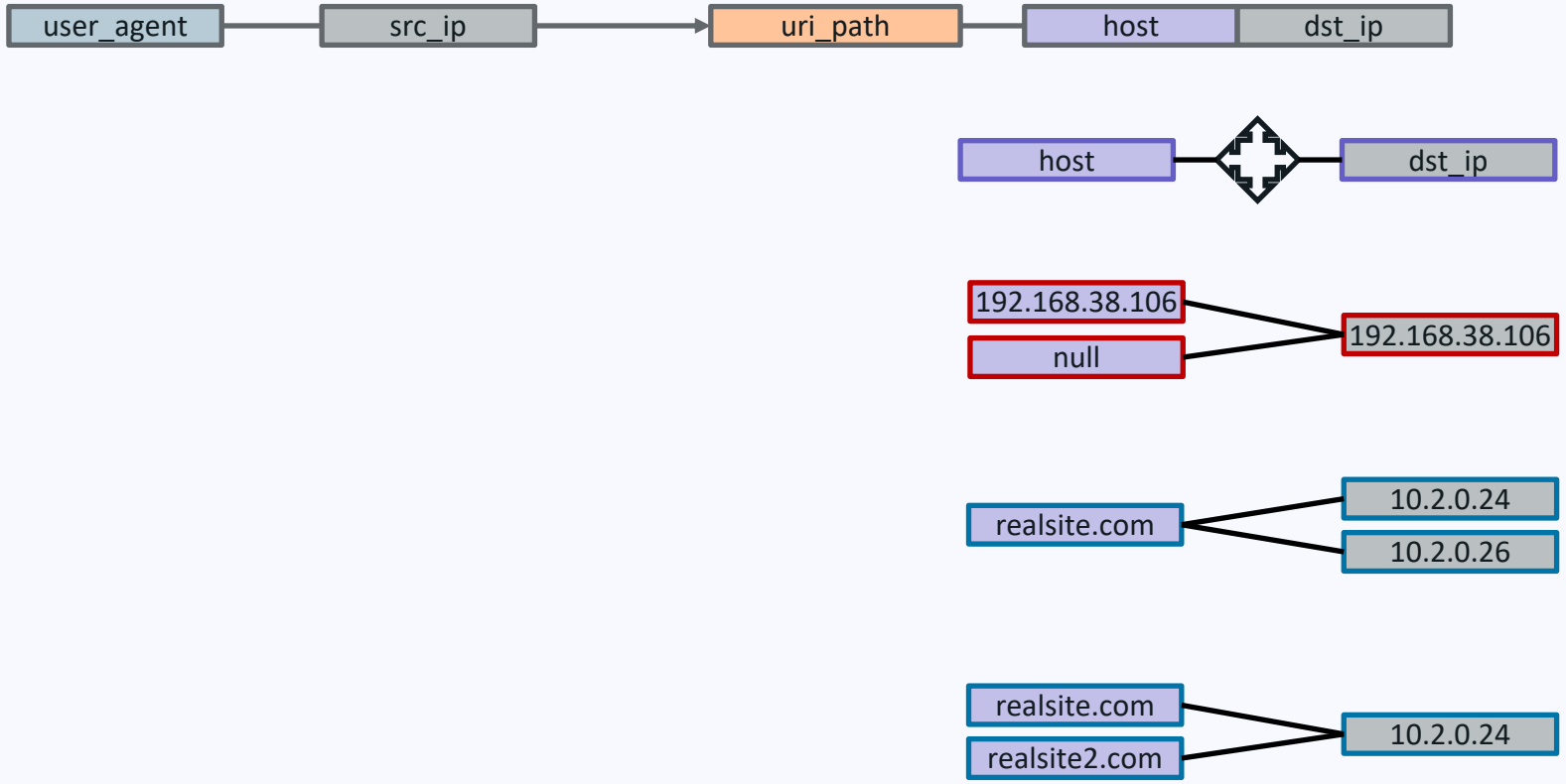
Finding the Object

Issues with host and destination IP field pairs



Finding the Object

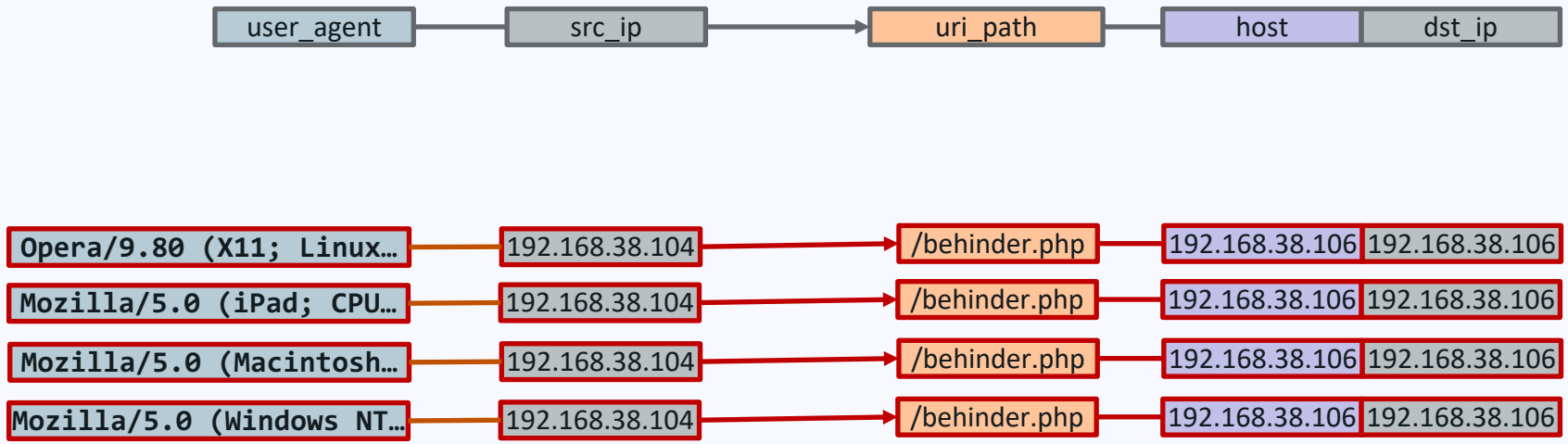
Issues with host and destination IP field pairs



= connected components

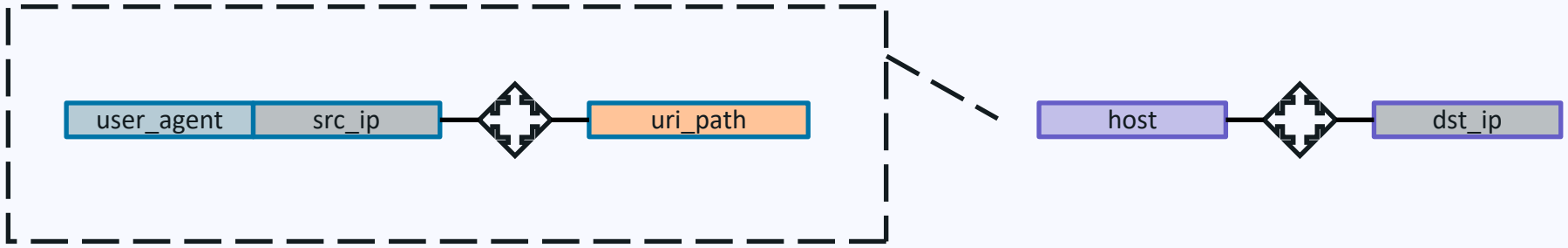
Finding the Object

Issues with multiple user agents



Finding the Object

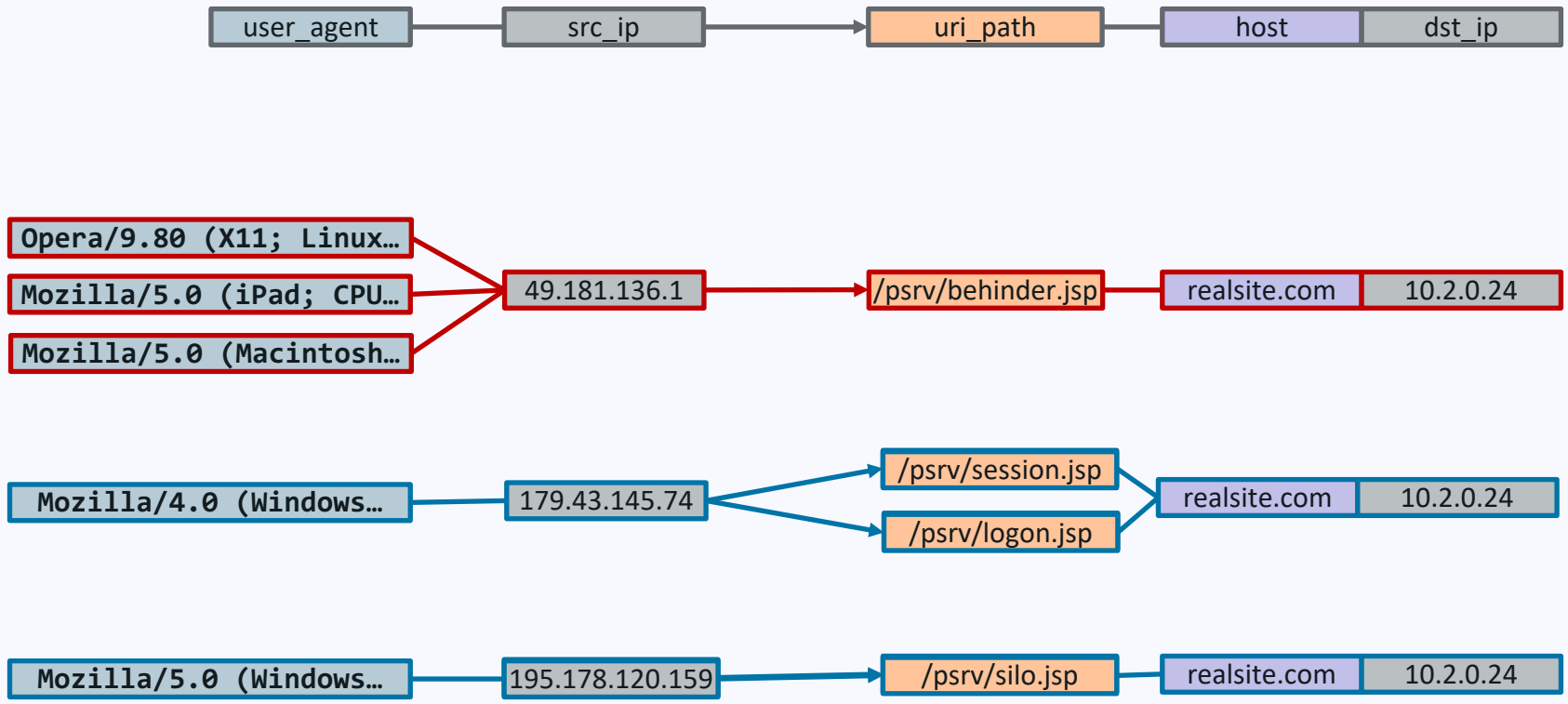
Final algorithm



= connected components

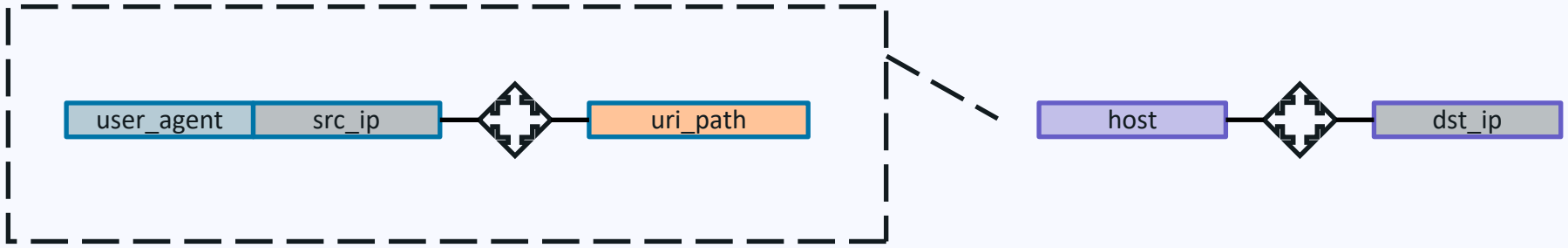
Finding the Object

Final algorithm example



Finding the Object

Final algorithm



= connected components

Dealing with Noise

- Specific noise for your environment
  - Possible example: Requests from internal addresses, but they look external since they go through a proxy
- Heuristic / Pattern
  - Scanner patterns: consistent User-Agent, source IP, uri_path with many destination IPs and failed connections
- Signature
  - Detection of bot user agents via regex list
    - E.g.: Pinterestbot, Baiduspider, YandexBot, MoodleBot

Tip:

Signature-based approaches to feature generation work better for noise than signal

Attackers may try to adapt to avoid detection, but noise sources don't care

Hard Cuts

- Expert decision
  - Wrong decision can result in FP
- When trying to reduce false positives, having a smaller set of items to start with helps
  - Reduces data set imbalance
- Can make hard cuts soft again

One webshell used the hard-coded User-Agent of:
Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)

Feature Finding Strategy

Iterative, not Comprehensive

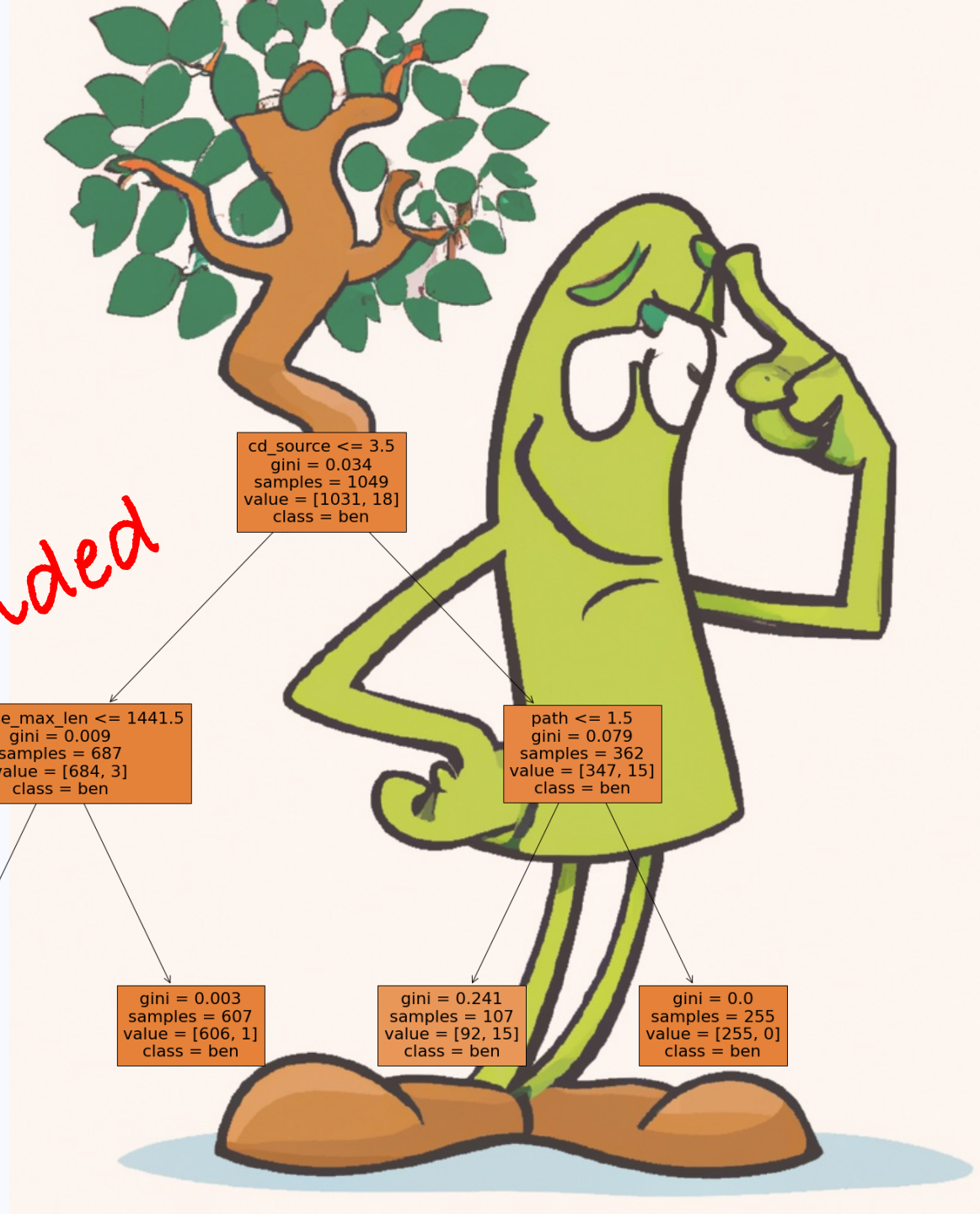
Strategy 1

- Think of all possible features
- Optimize model

Strategy 2

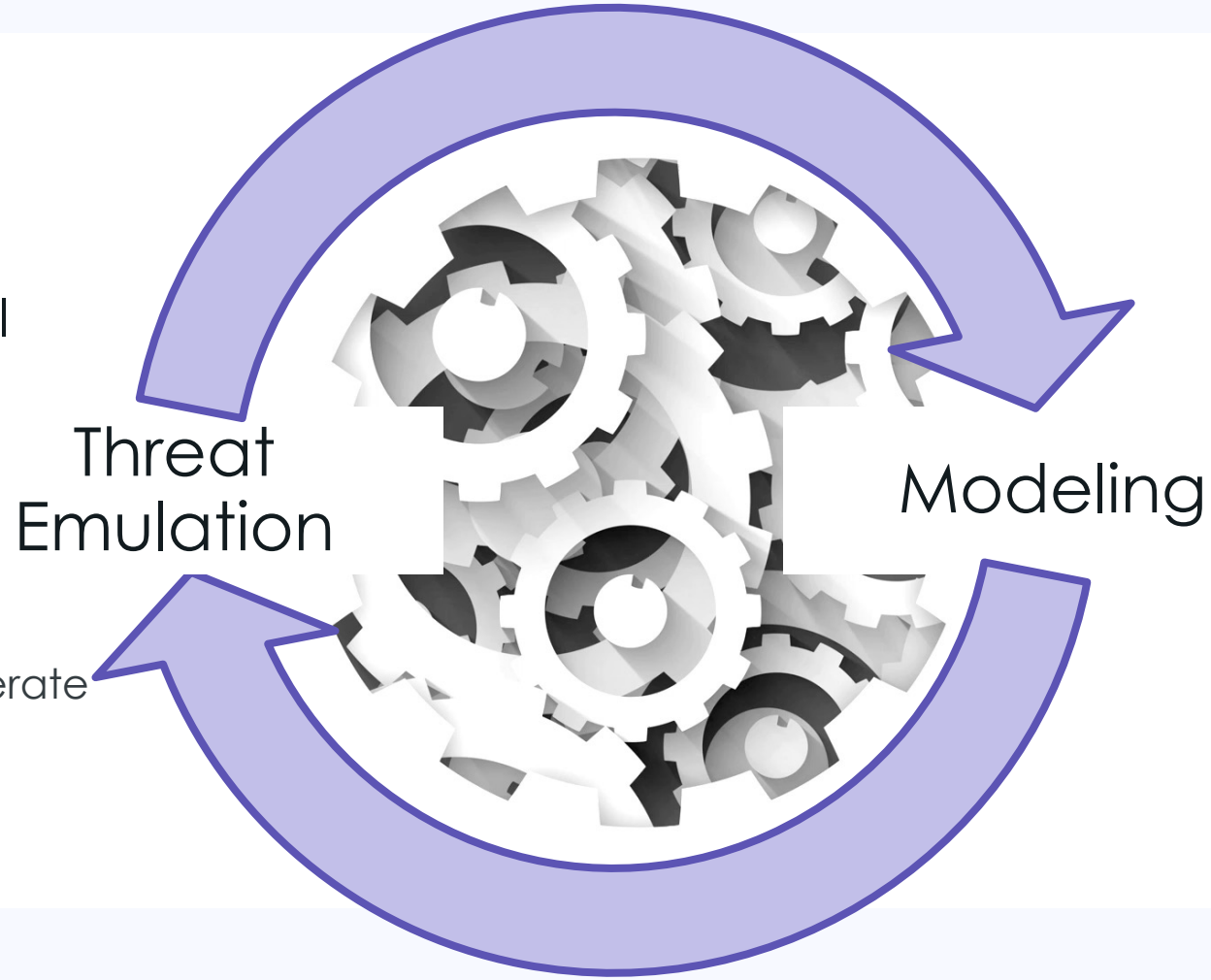
- Think of a few features, apply hard cuts
- Evaluate remaining subsets to identify additional features
- Repeat

Recommended



Threat Emulation

- Started with 8 manufactured web shell episodes
  - Behinder, ASPXSpy, Godzilla, antsword, etc
- Early model was too effective
  - Recognized that one of the key features corresponded with the process used to generate examples
- Ultimately generated 18 episodes
  - Increased variability of attacker behavior in additional episodes
  - Made problem more difficult

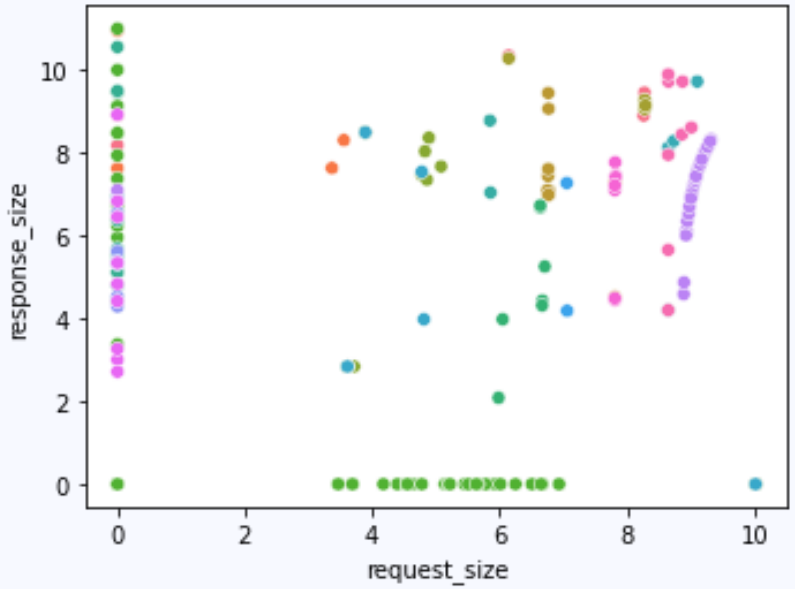


© 2017-2022 Gigamon Inc. All rights reserved.

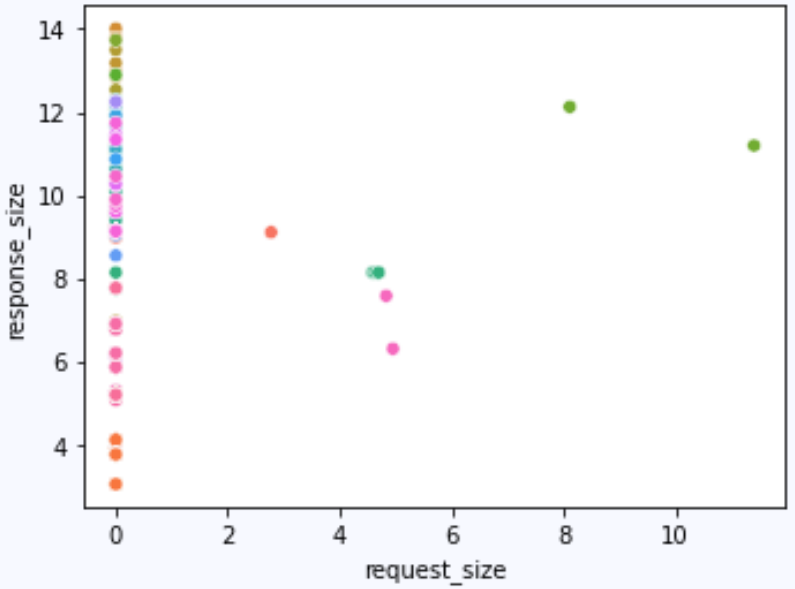
Modeling

Strong features

Malicious

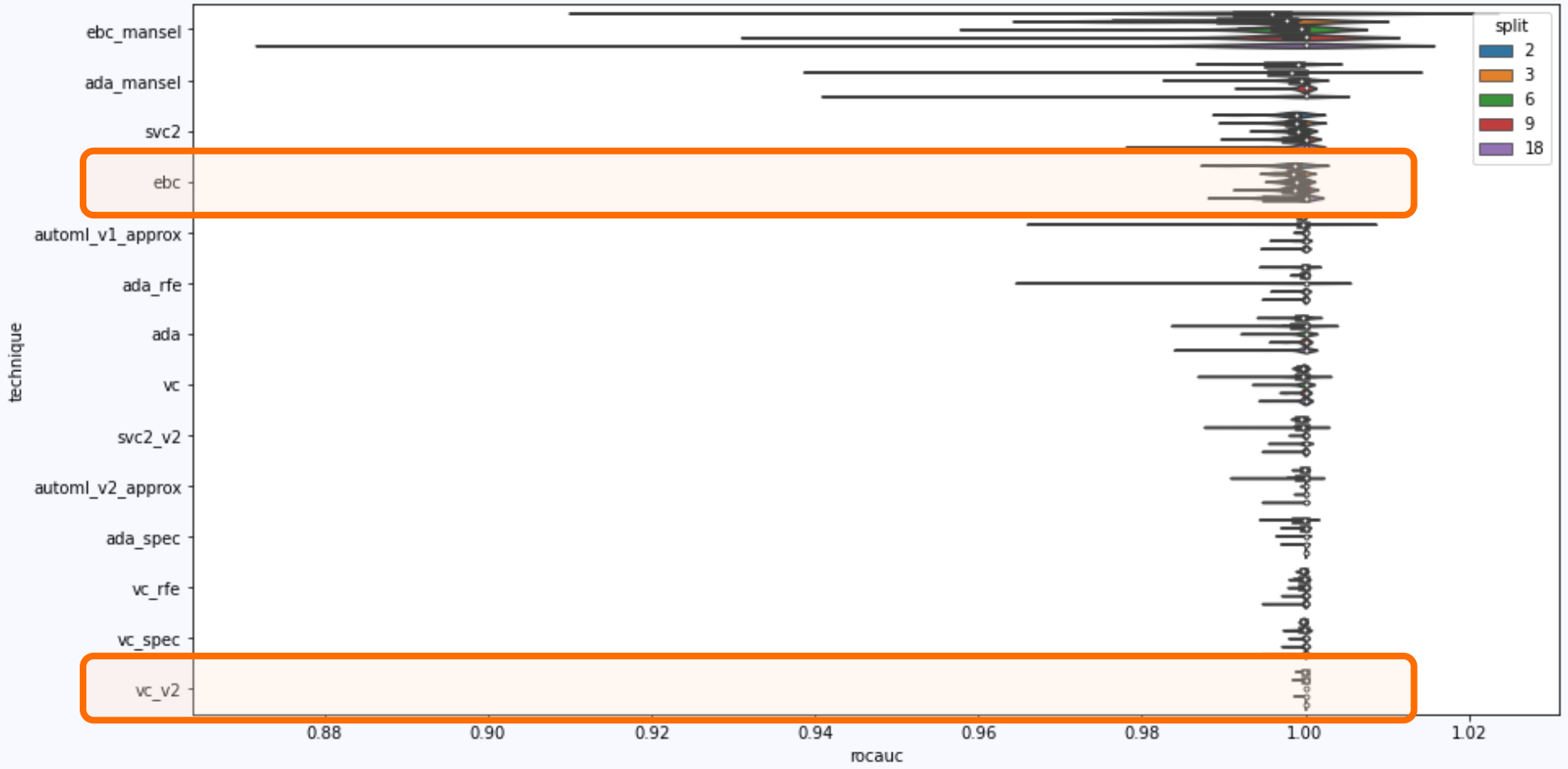


Background



Modeling

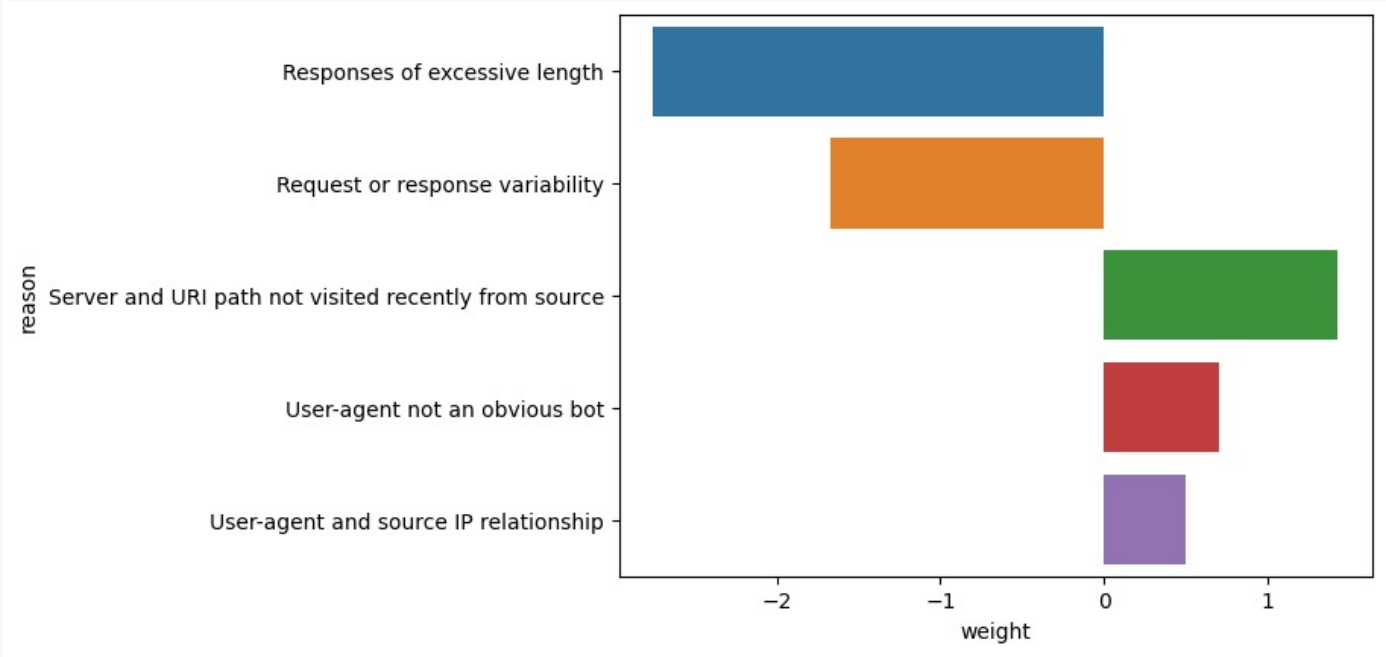
Finding a suitable model



Modeling

Using multiple models

- High performing
  - Ensemble of Support Vector Machine (with optimized parameters) and AdaBoosted trees
  - Effective for generalized detection with low FP rate
- Explainable
  - ExplainableBoostingClassifier
  - Increased chance of FPs, but useful for hunting and provides context
  - Specifically select features for:
    - Independence (lack of correlation)
    - Understandability
    - Value to model

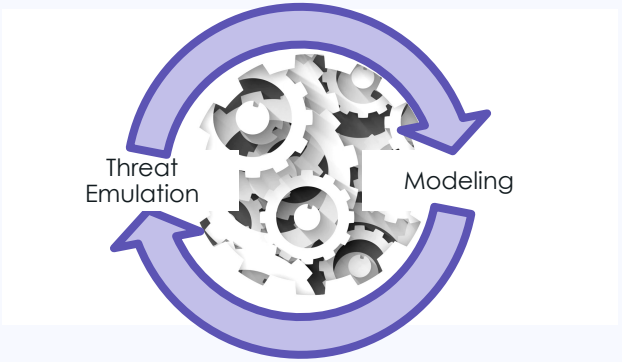


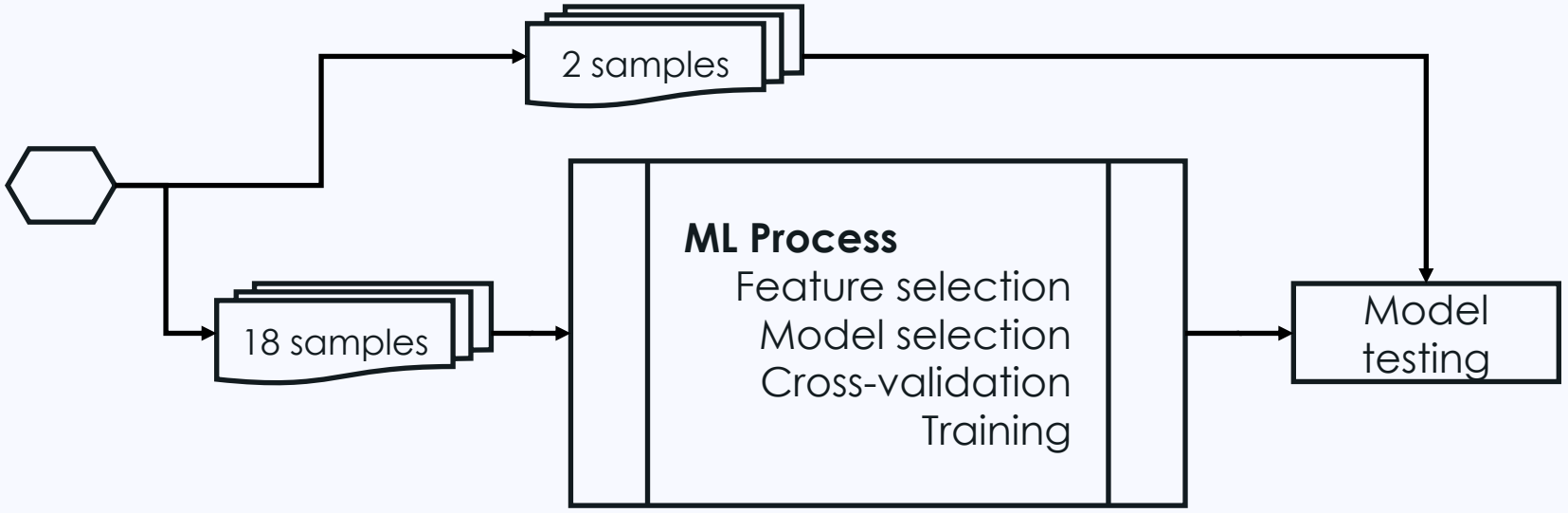
Summarized reasons for prediction

(Negative sample, negative prediction)

Meta Validation

- Embedded validation into process, not just model creation





© 2017-2022 Gigamon Inc. All rights reserved.

10 secrets for creating a crazy good network detection model

(Do these and you'll put Suricata signature writers out of a job)

- Overlay emulated malicious traffic over real traffic
- Use high-level protocol information (Zeek)
- Focus detection scope
- Maintain feedback loop between emulation and modeling
- Carefully define object of focus
- Remove noise (with signatures if needed)
- Make hard cuts (prudently)
- Be the decision tree (to find features)
- Explain your work
- Validate outside the process


Questions