Detecting Unknown Encrypted Malicious Traffic in Real Time via Flow Interaction Graph Analysis

Effective and Efficient Detection for Encrypted Malicious Traffic

Chuanpu Fu[1], Qi Li[1,2], Ke Xu[1,2]

[1]Tsinghua University, Beijing, China; [2]Zhongguancun Lab, China

1. Backgrounds: Traffic Encryption

- Traffic encryption is widely adopted on the Internet.

May 2019, 94% of all Google web traffic is encrypted.[1]

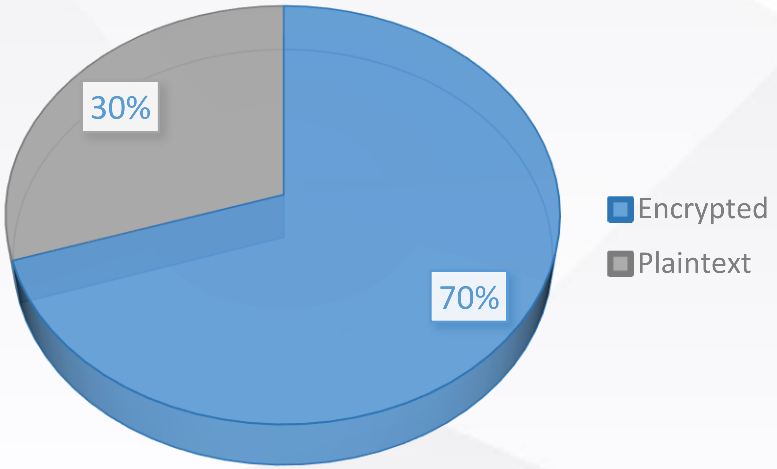
Nearly 80% of web pages loaded by Firefox use HTTPS.[2]

Over 98% Alexa top 1k websites support HTTPS.

[1] https://transparencyreport.google.com/https/overview?hl=en
[2] Predicts 2017: Network and Gateway Security.

1. Backgrounds: Abused Traffic Encryption

- Traffic Encryption is double-edged.
  - Attackers abuse traffic encryption to conceal their behaviors, e.g., data breach, and exfiltration.
  - It is reported that, 70% attacks were constructed by encrypted traffic in 2020.



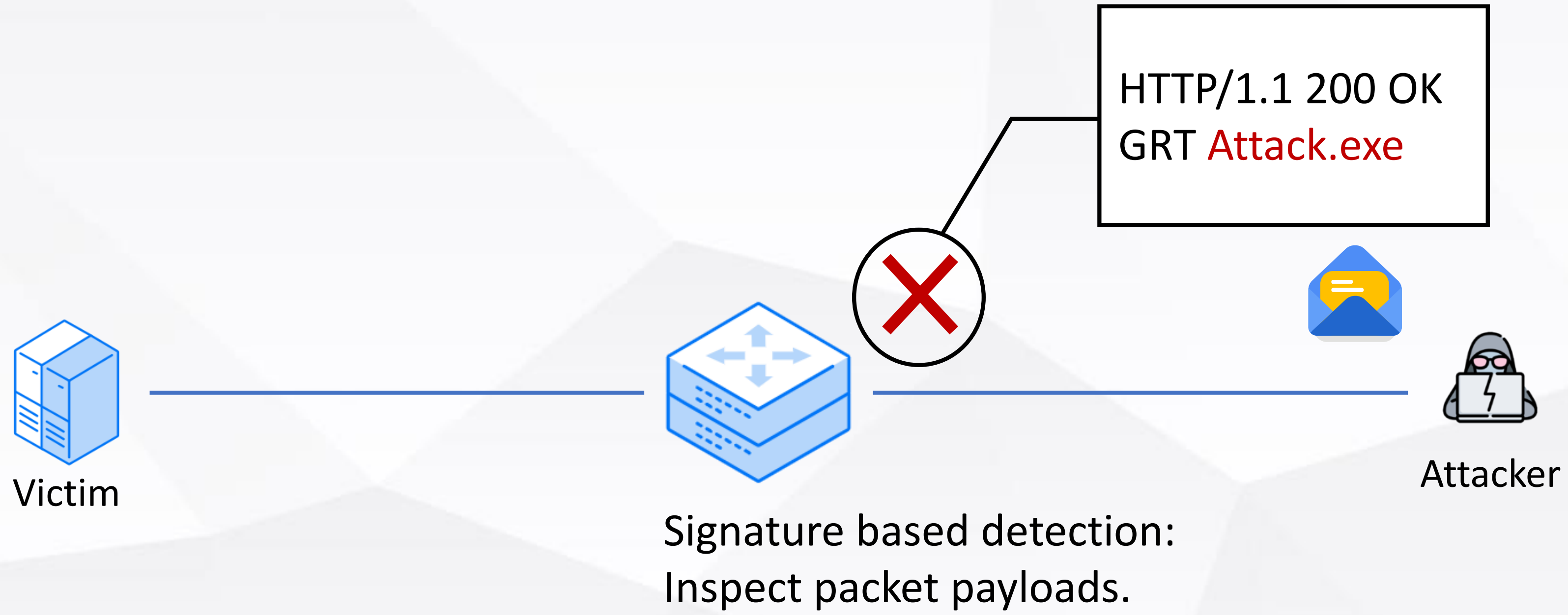
Over 70% attacks were constructed by encrypted attack traffic.

[3] Cisco Encrypted Traffic Analytics White Paper, Cisco.

1. Backgrounds: Malicious Traffic Detection

- Attackers can easily evade the existing detection via traffic encryption.

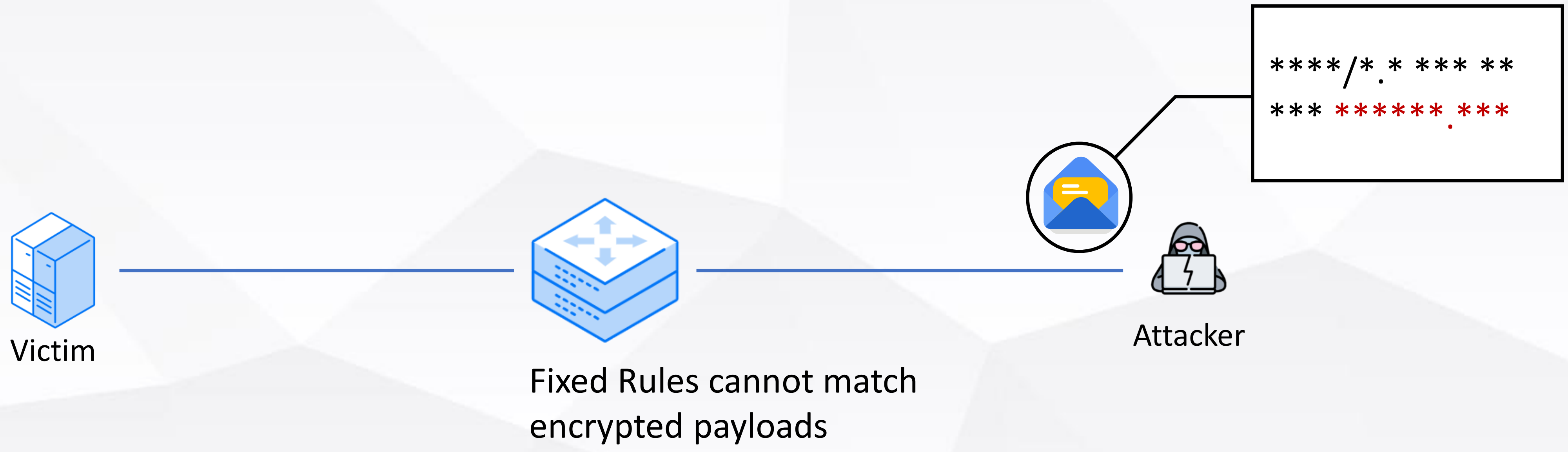
Traditional signature-based method:



1. Backgrounds: Malicious Traffic Detection

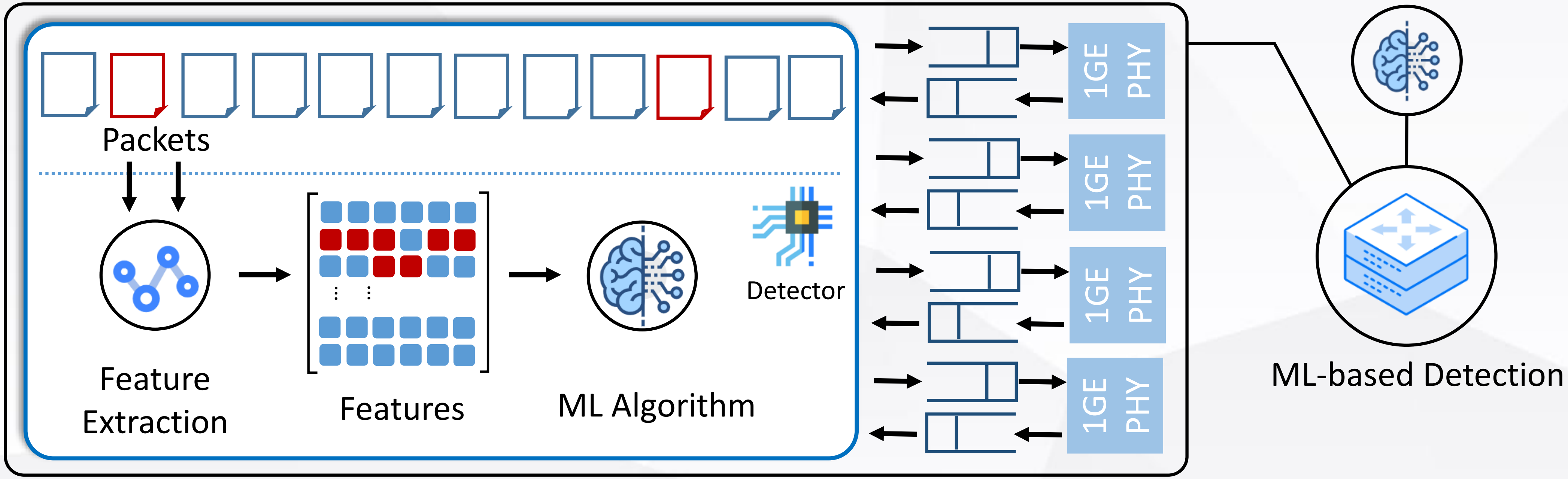
- Attackers can easily evade the existing detection via traffic encryption.

  Traditional signature-based method: Deep Packet Inspection (DPI) is invalid.



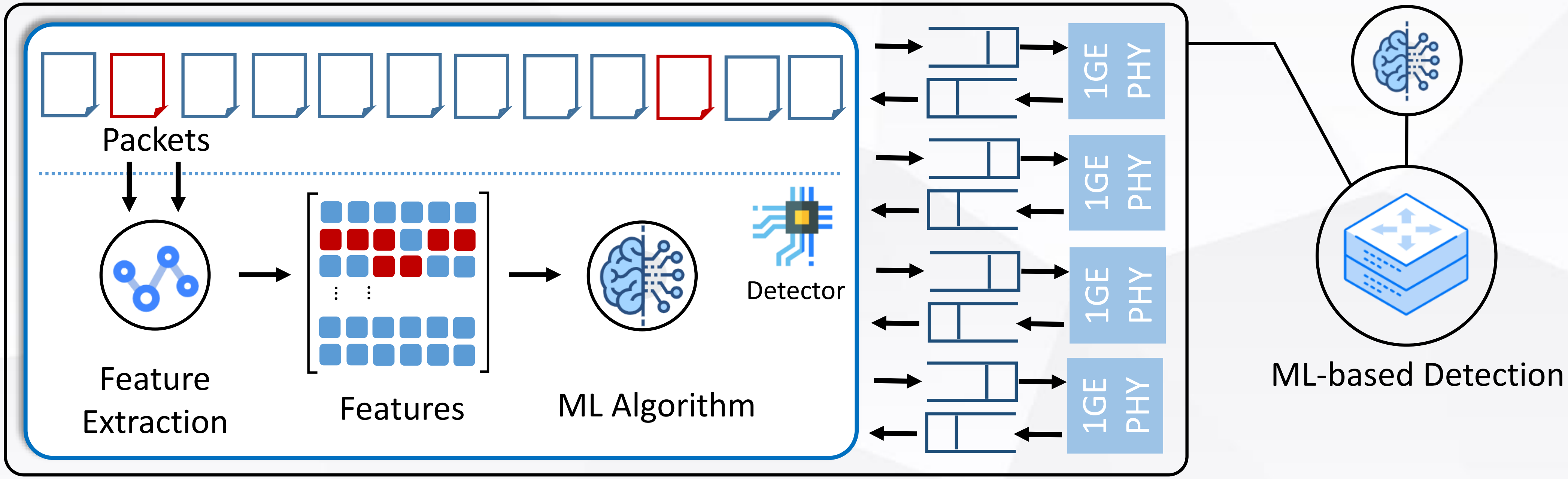
1. Backgrounds: Malicious Traffic Detection

- Attackers can easily evade the existing detection via traffic encryption.



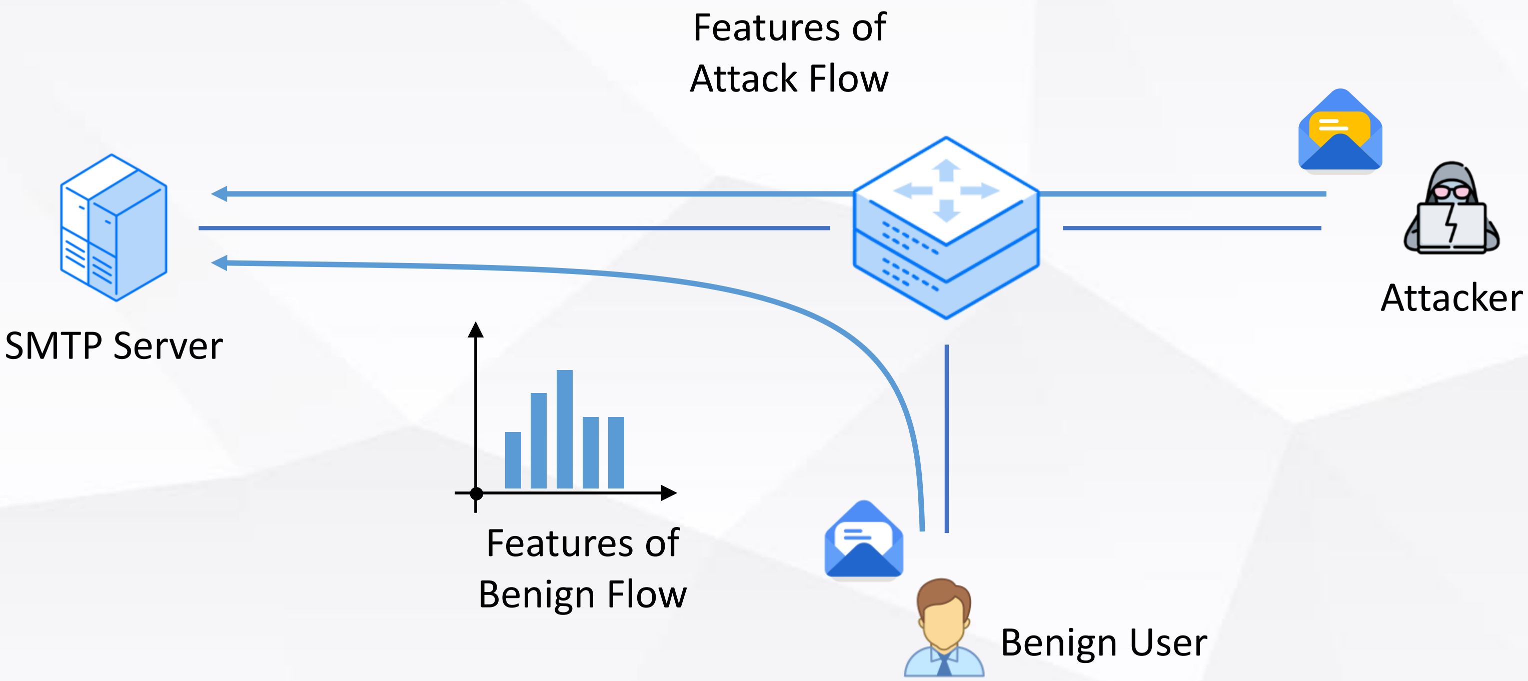
1. Backgrounds: Malicious Traffic Detection

- Attackers can easily evade the existing detection via traffic encryption.
- Advanced ML-based detection cannot detect such attack either.
  - *Encrypted malicious flows with benign traffic patterns.*



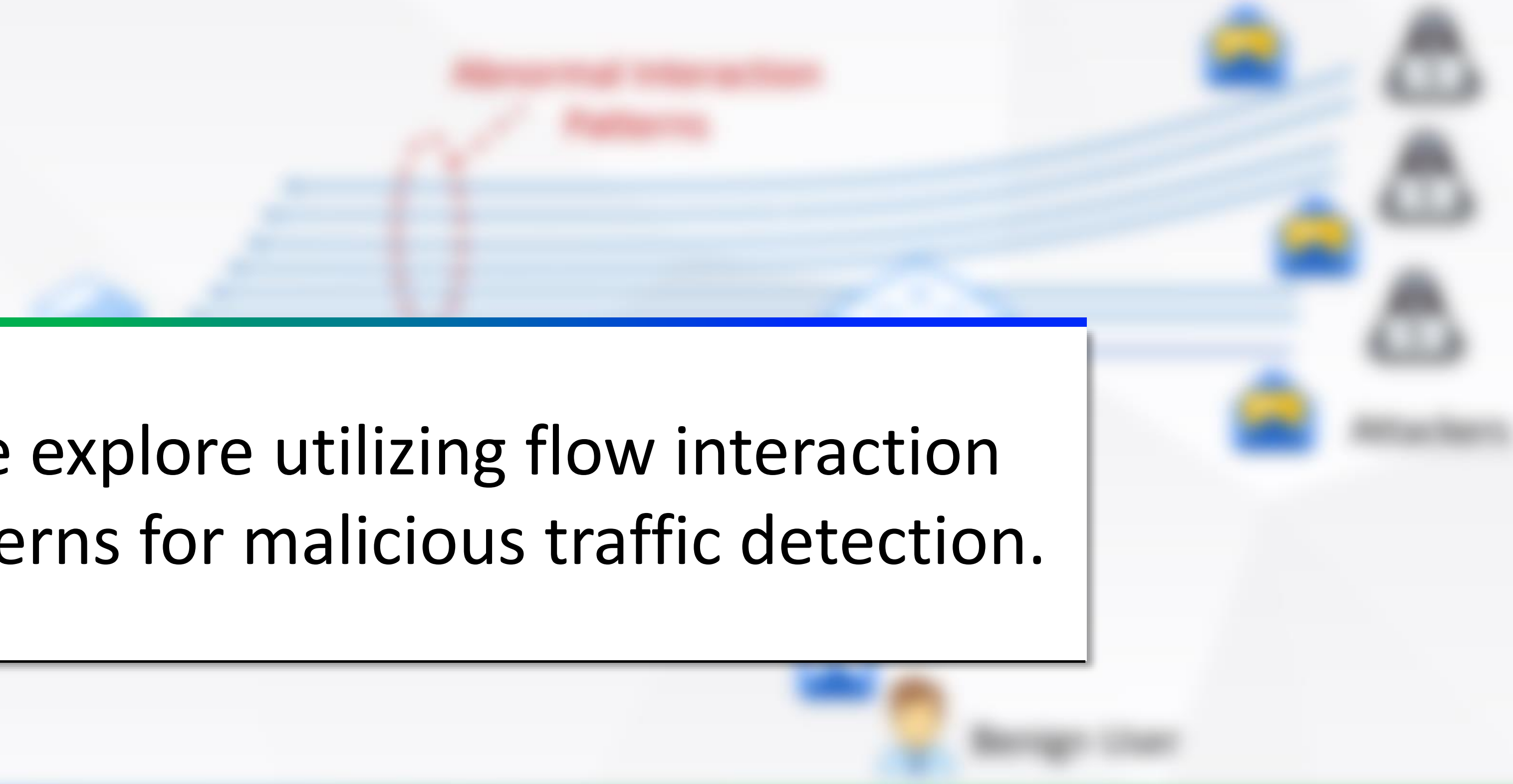
1. Backgrounds: Encrypted Attack Traffic Evades Detection

- Advanced ML-based detection cannot detect such attack either.
  - Benign SMTP-over-TLS Traffic & Encrypted Spam Traffic.
  - Traditional traffic features cannot differentiate encrypted malicious traffic.



2. Motivation: Interaction Patterns

- It is still possible to detect encrypted malicious traffic according to interaction patterns.
  - The interactions between spambots and SMTP servers are significantly frequent.



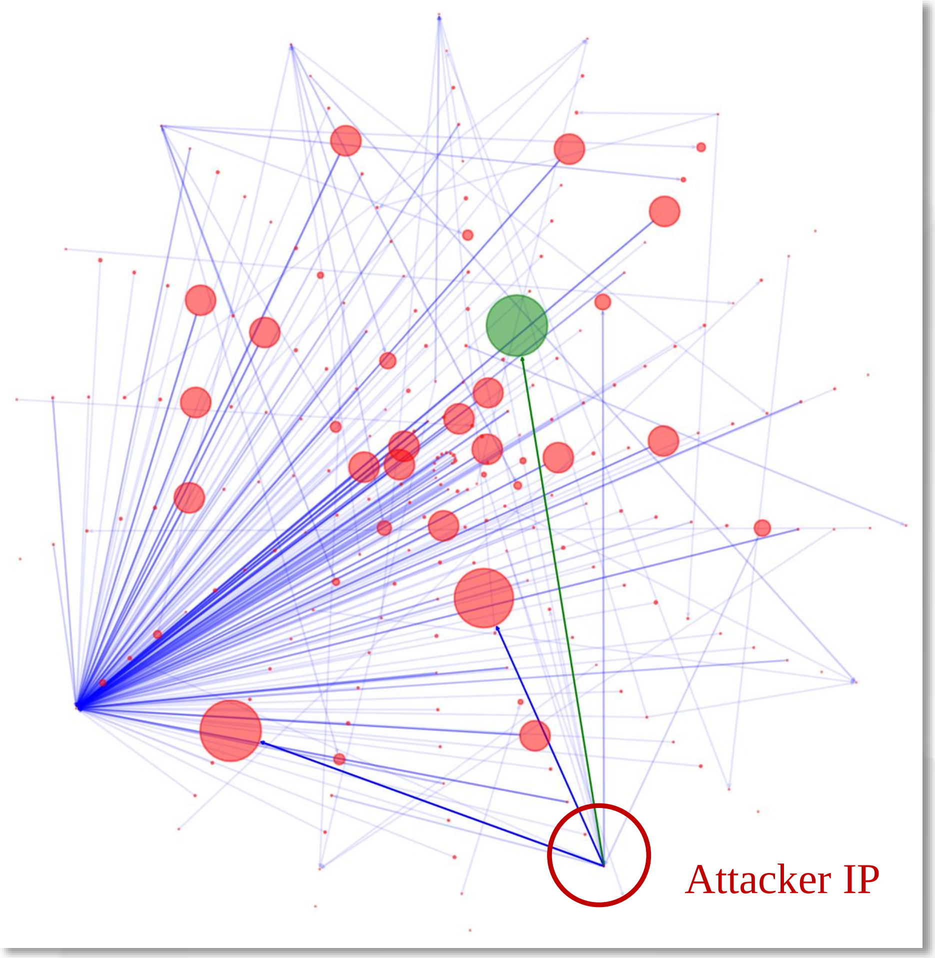
We explore utilizing flow interaction patterns for malicious traffic detection.

2. Motivation: Flow Interaction Graph

- We use a graph to represent the interaction patterns.
  - Vertices → IP addresses.
  - Edges → Flows

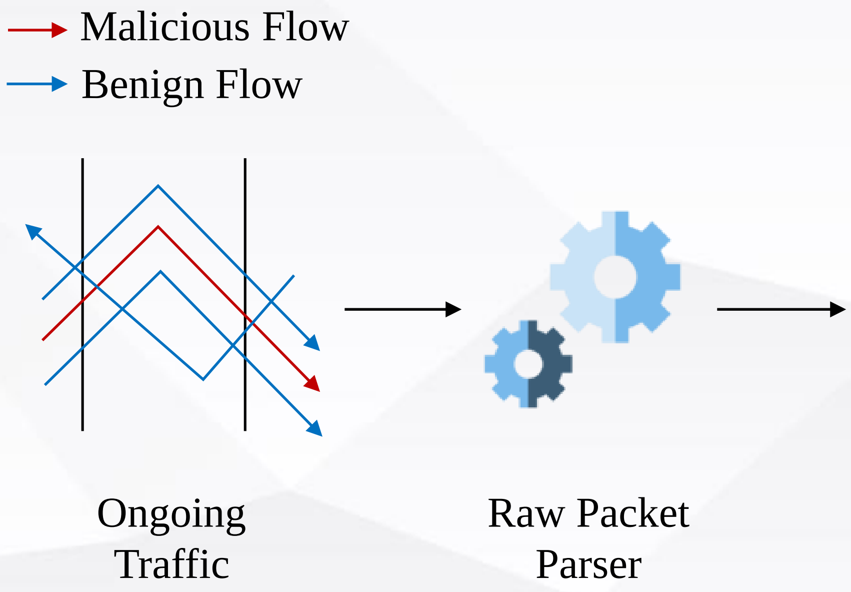
- We use unsupervised graph learning to detect the attacks, without requiring any prior knowledge.

Flow Interaction Graph



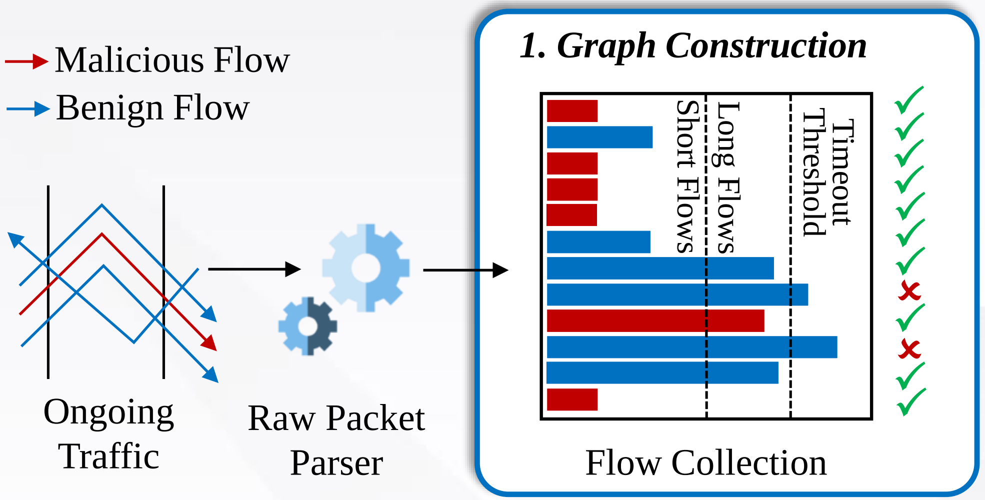
3. Design: Overview

➢ Module 1: Graph Construction Module.



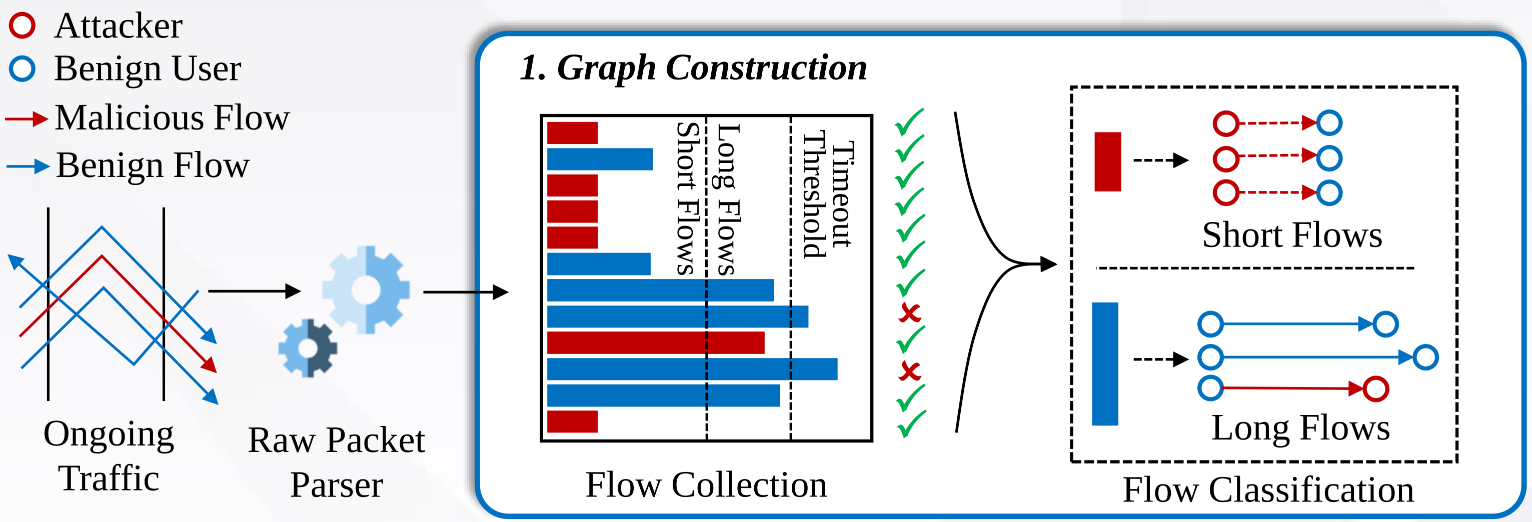
3. Design: Overview

- Module 1: Graph Construction Module.



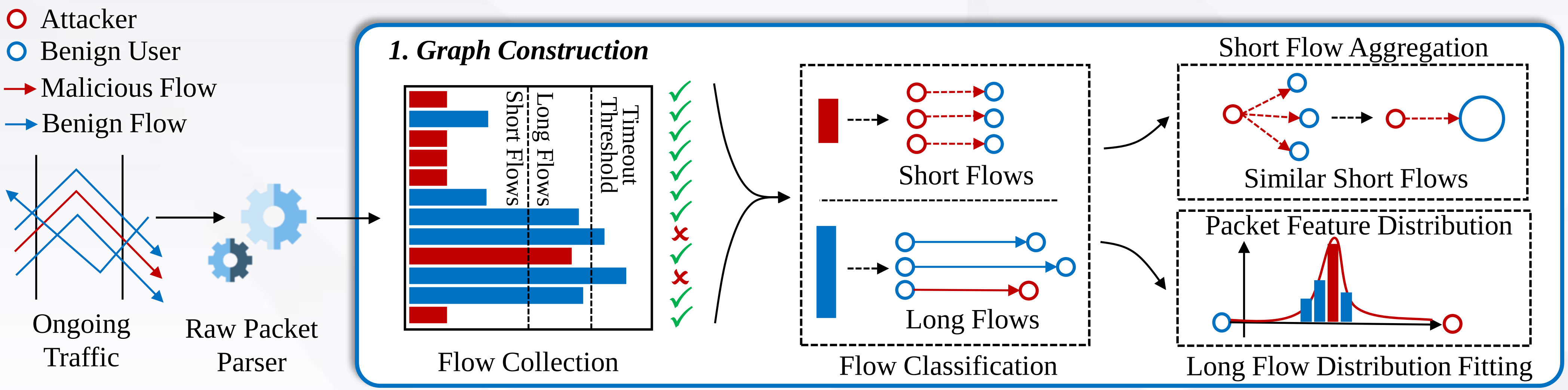
3. Design: Overview

➢ Module 1: Graph Construction Module.



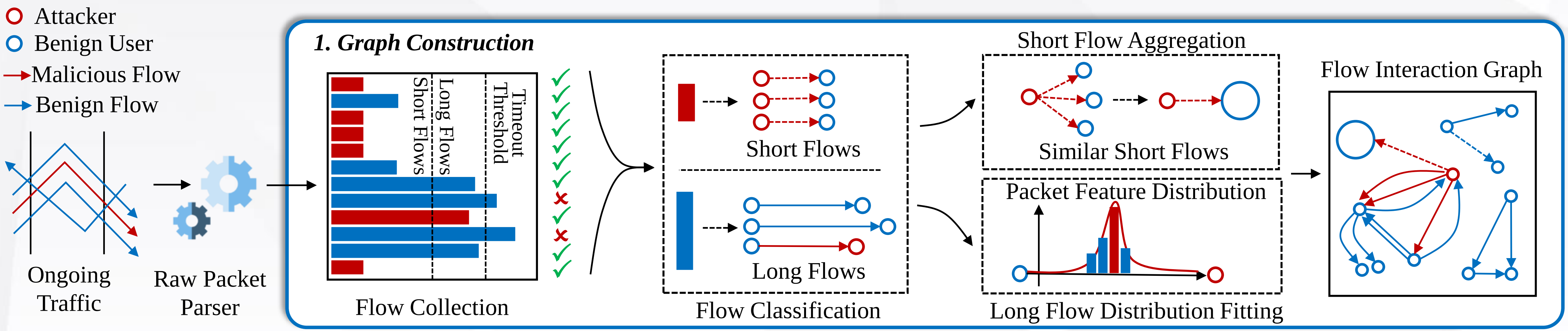
3. Design: Overview

➢ Module 1: Graph Construction Module.



3. Design: Overview

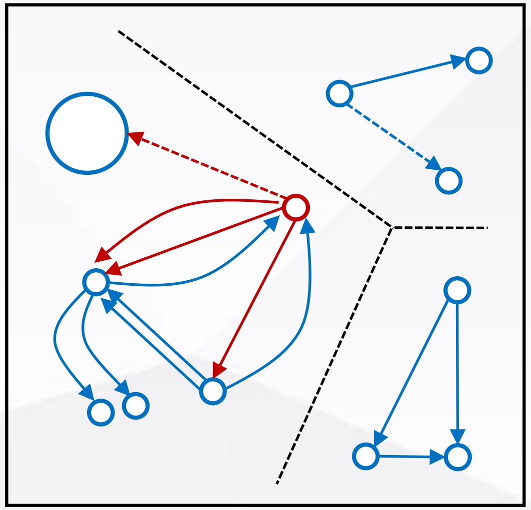
➢ Module 1: Graph Construction Module.



3. Design: Overview

- Module 2: Graph Pre-Processing Module.

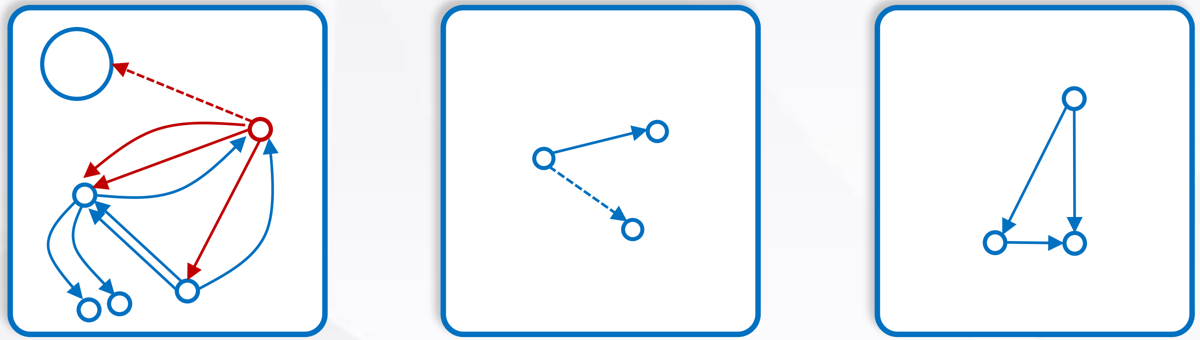
Flow Interaction Graph



3. Design: Overview

- Module 2: Graph Pre-Processing Module.

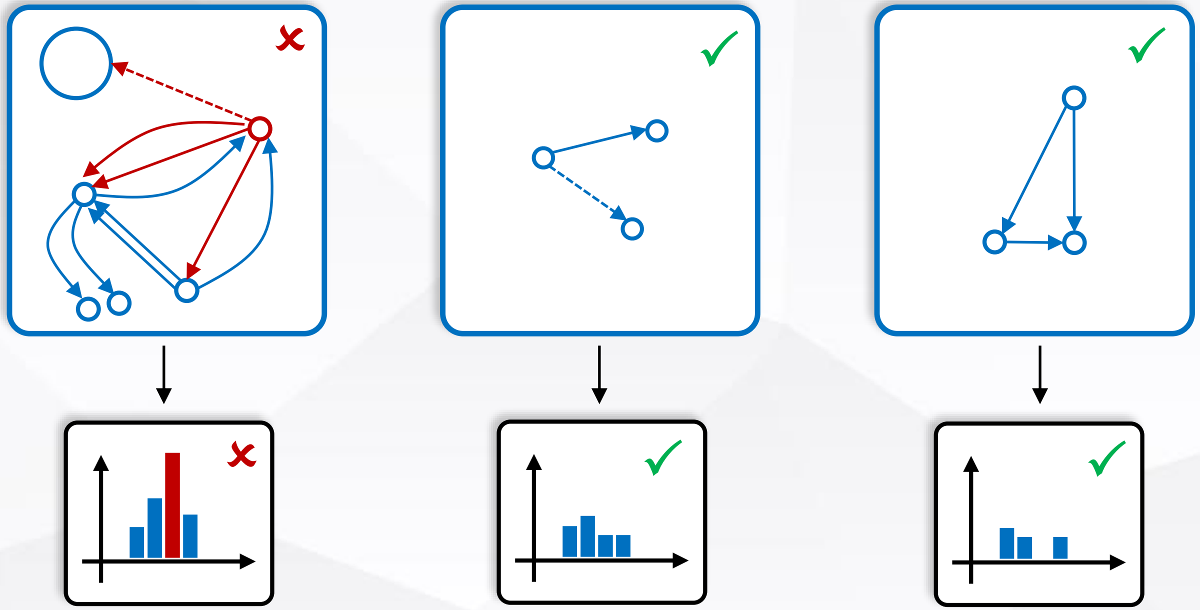
Strongly Connected Components



3. Design: Overview

- Module 2: Graph Pre-Processing Module.

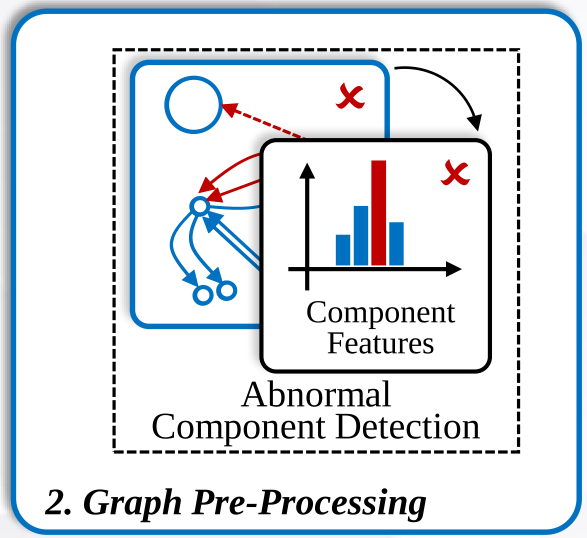
Strongly Connected Components



Component Statical Features

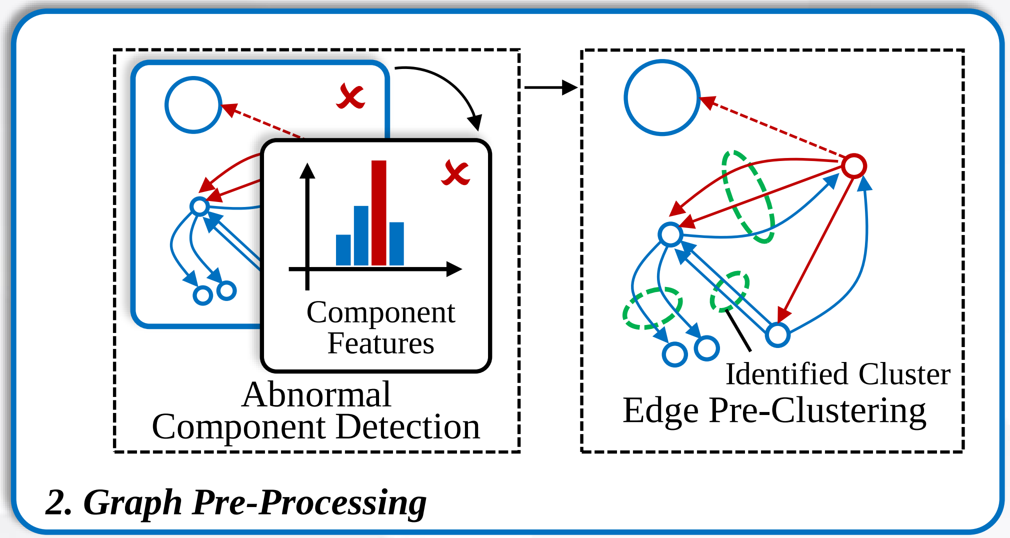
3. Design: Overview

- Module 2: Graph Pre-Processing Module.



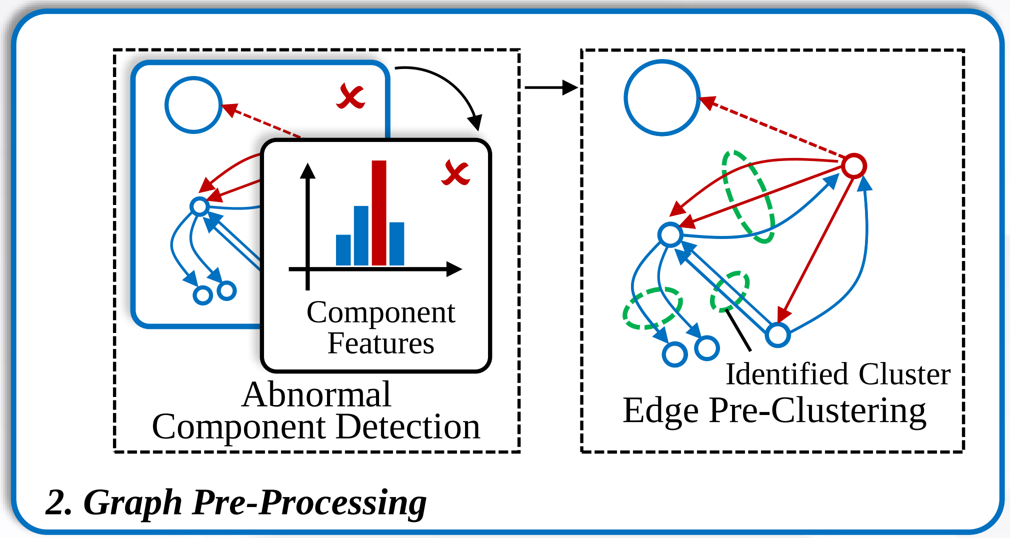
3. Design: Overview

- Module 2: Graph Pre-Processing Module.



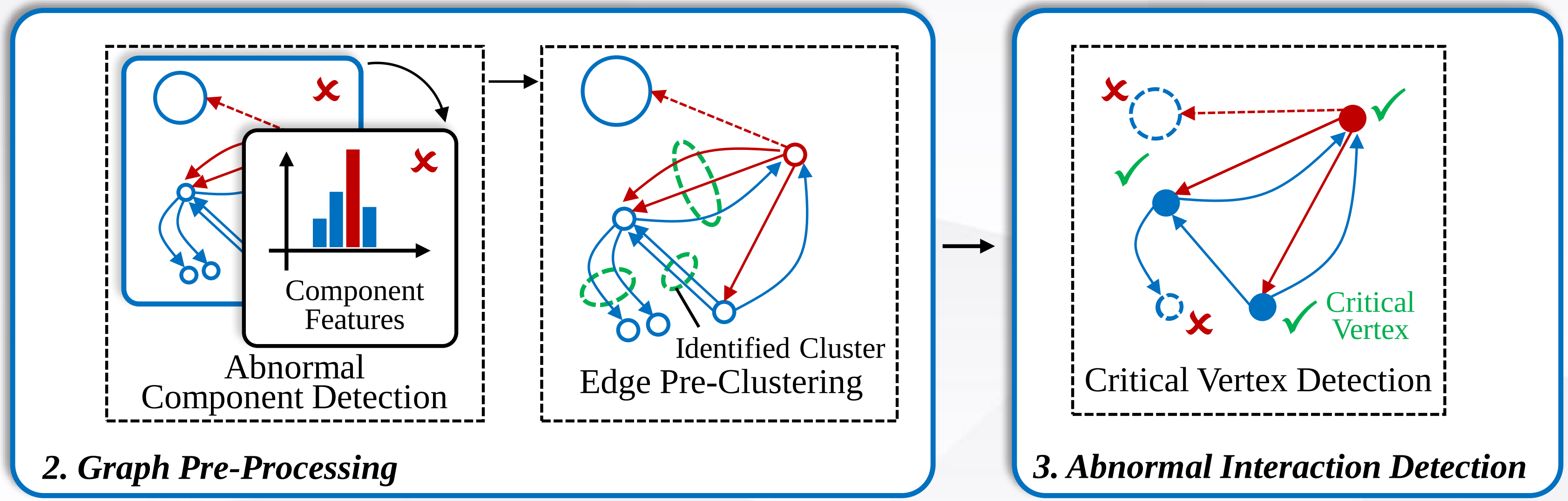
3. Design: Overview

- Module 2: Graph Pre-Processing Module.



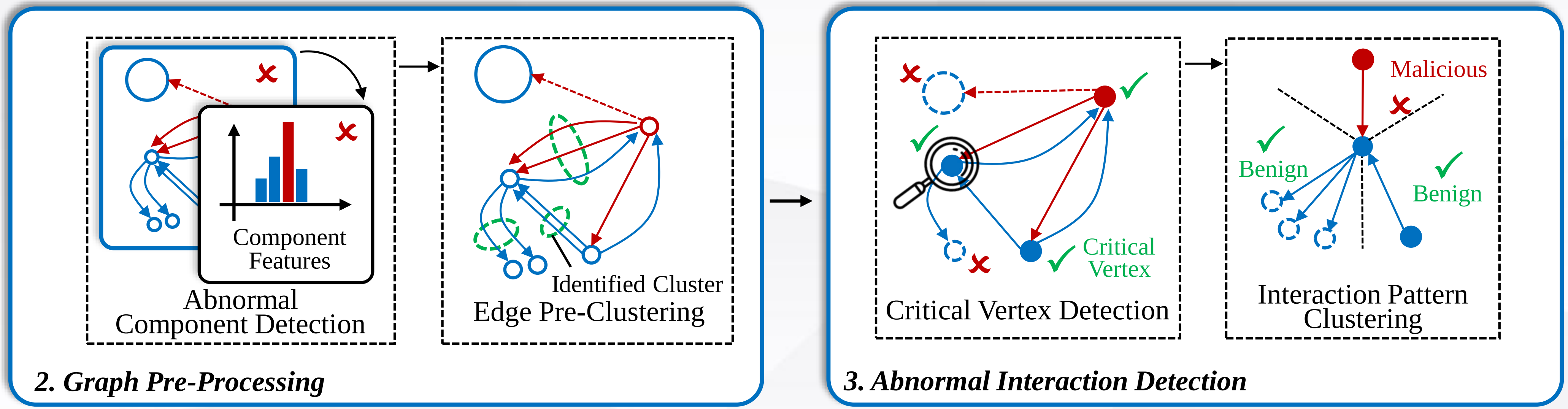
3. Design: Overview

- Module 3: Graph Detection Module.



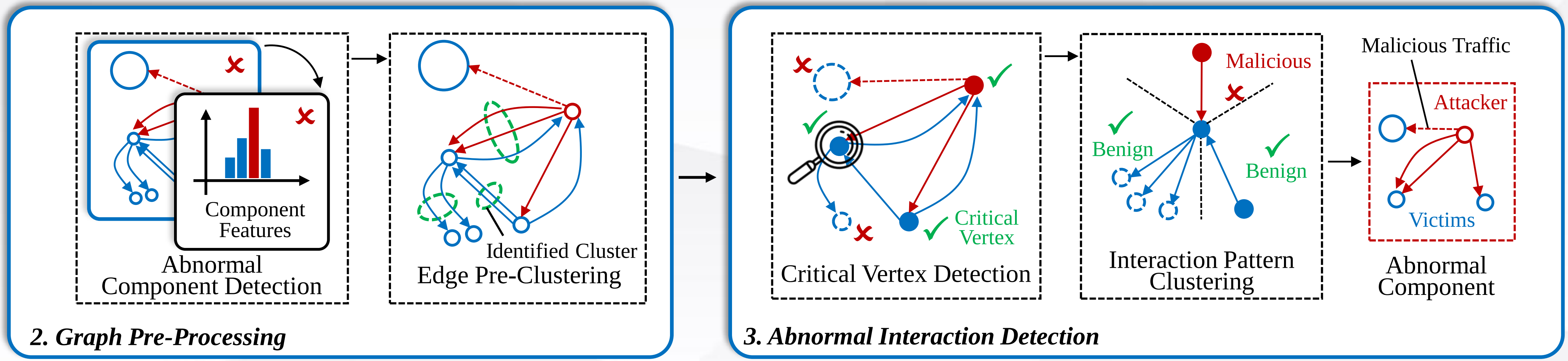
3. Design: Overview

- Module 3: Graph Detection Module.

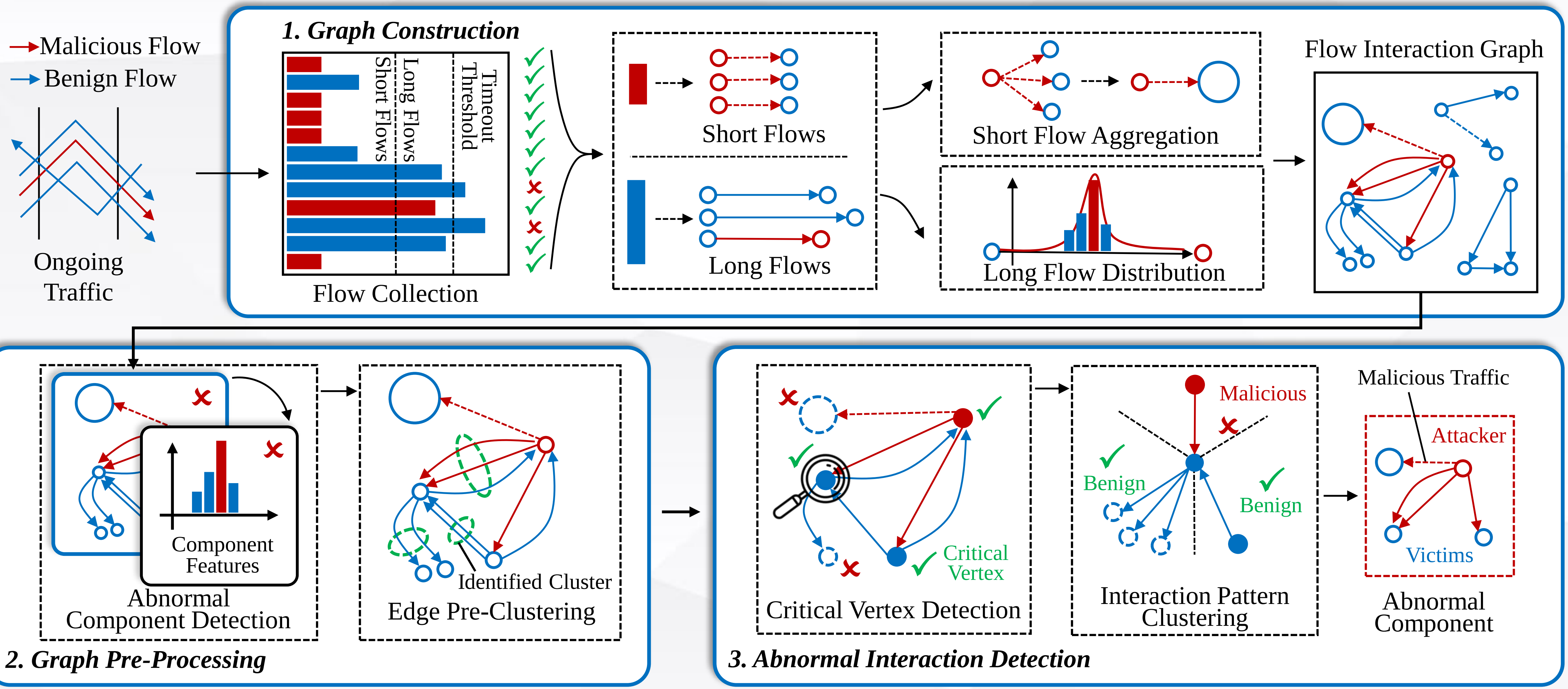


3. Design: Overview

- Module 3: Graph Detection Module.



3. Design: Overview



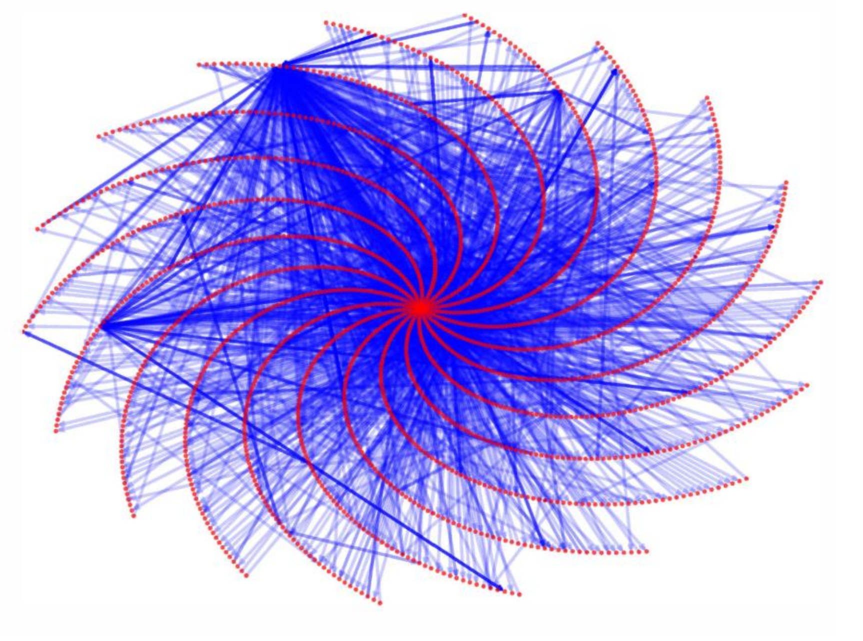
3. Design: How to reduce graph density?

- Complex flow interaction patterns.
  1. Over 50,000 active hosts reside in AS2500.
  2. Over 3M flows per hour.

- We cannot use one edge to denote one flow and use one vertex to denote one IP → *dependency explosion problem*.

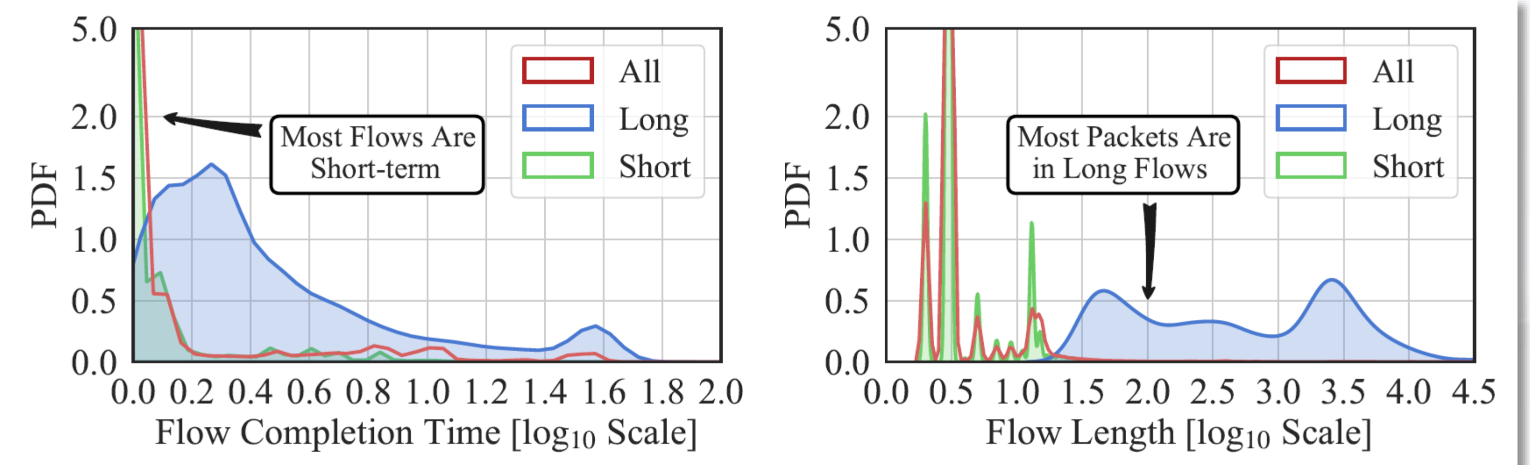
  - How to reduce the density of a graph?

Graph Before Density Reduction



3. Design: How to reduce the graph density?

- Observation: most flows are short flow, and most packets are in long flow.
- Solution: we construct edges to represent short and long flow, separately.



(a) FCT distribution. (b) Flow length distribution.

3. Design: How to reduce the dense graph?

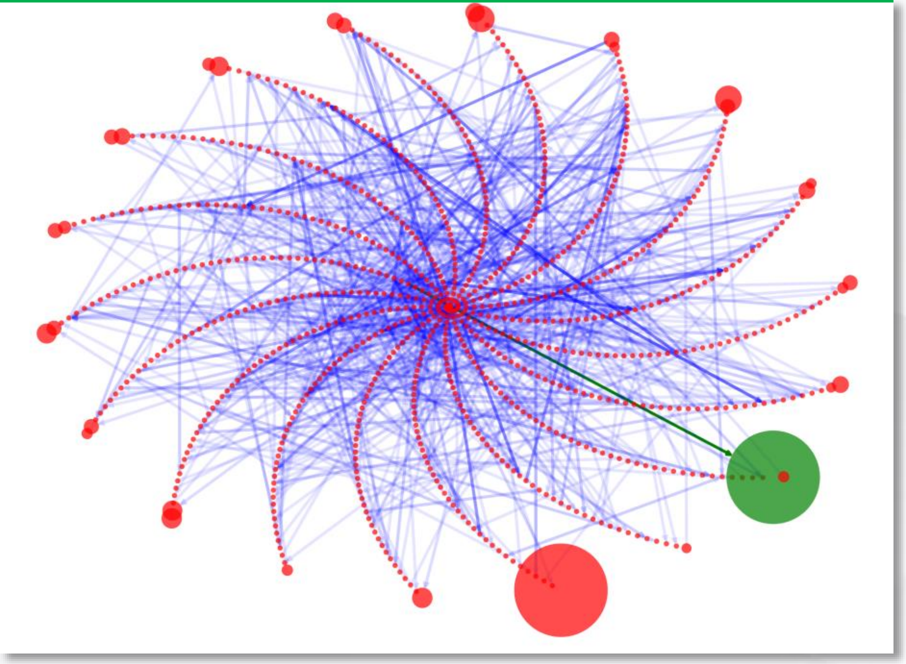
- Many short flows are similar, e.g., DNS queries, password cracking.
  - We aggregate the short flows and use one edge to represent many short flows

- Long flows have complex patterns.
  - We extract fine-grained features for long flows, i.e., distribution features.

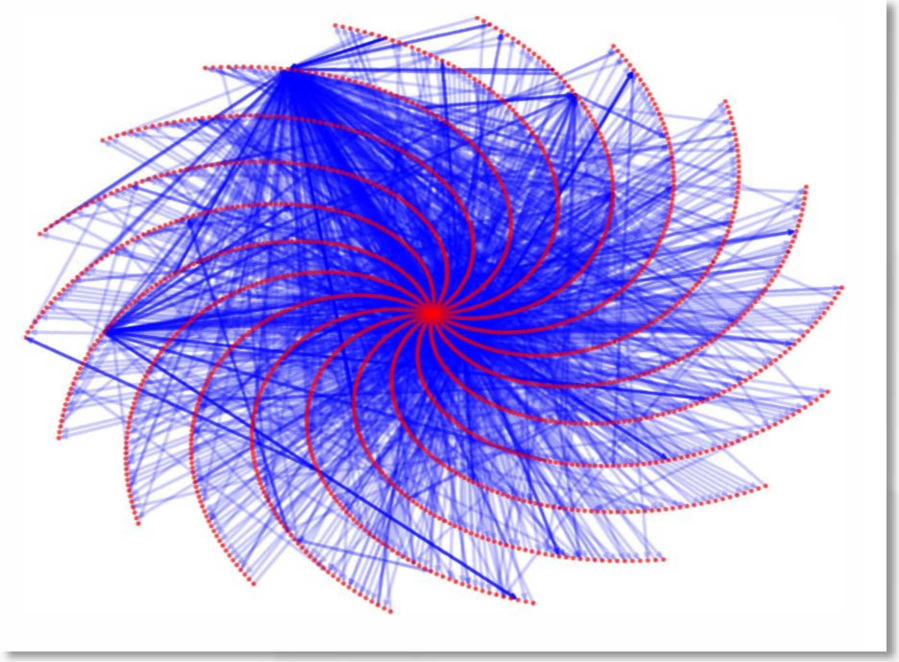
- One edge → many short flows or one long-flow.
- One vertex → a group of addresses or one address.

3. Design: How to reduce the dense graph?

Graph After Density Reduction



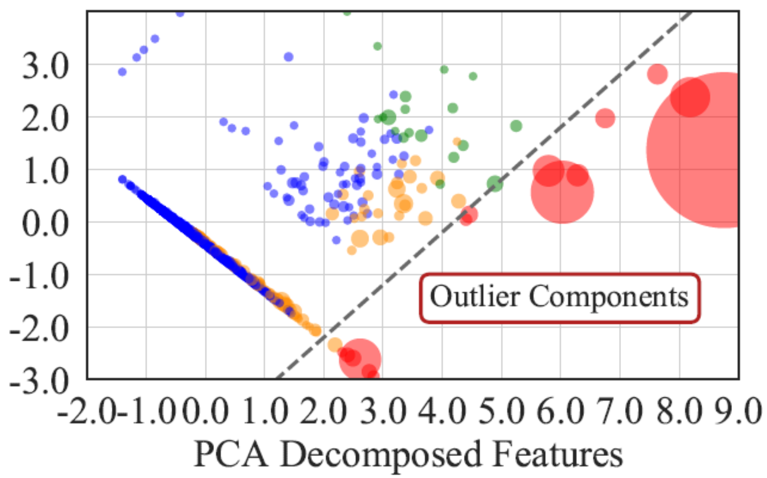
Graph Before Density Reduction



3. Design: How to efficiently identify attack traffic?

- The size of graph is still too large for real-time graph learning.

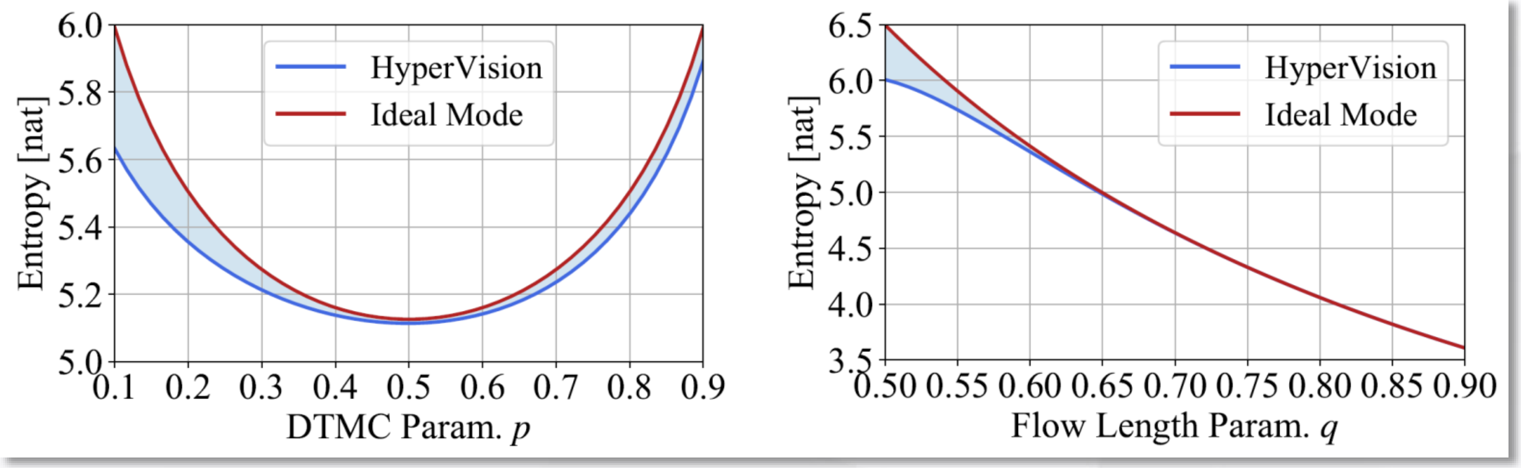
- We exclude benign components by clustering the high-level statistics.



(b) Scatter of the components.

4. Theoretical Analysis

- To prove the effectiveness of the method, we developed an information theory based analysis framework, which models flows by using DTMC.
- By calculating the entropy of the DTMC, we prove the amount of information preserved on the graph is near-optimal.



5. Experimental Analysis: Setup

- We implement our method using Intel DPDK (Data Plane Development Toolkit).
  - *The source code is publicly available.*

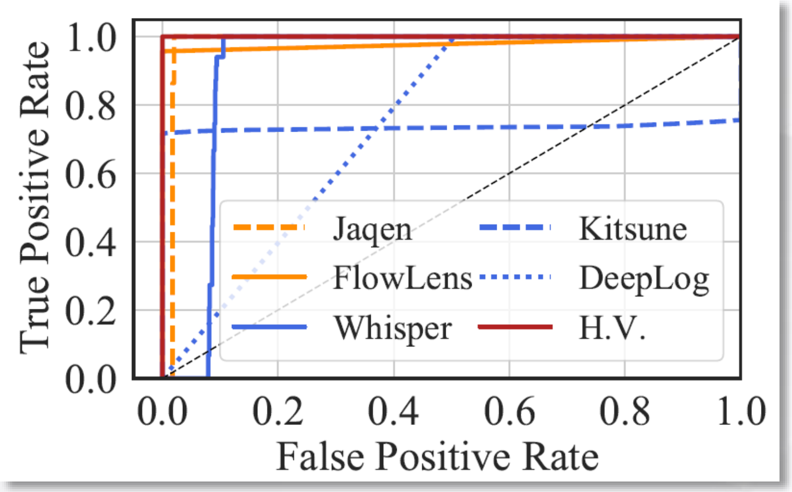
- On the physical testbed, we replay 92 kinds of malicious traffic, including 48 attacks with encrypted malicious traffic:
  - *Traditional brute attacks* (e.g., amplification attacks).
  - *Encrypted flooding traffic* (e.g., the Crossfire Attack).
  - *Encrypted Web attack traffic* (e.g., CVE-2013-2028).
  - *Malware generated traffic* (e.g., C&C Channel).

- These attack traffic is collected form a scaled private cloud network (> 1500 users), and the malware traffic is manually extracted form public datasets.

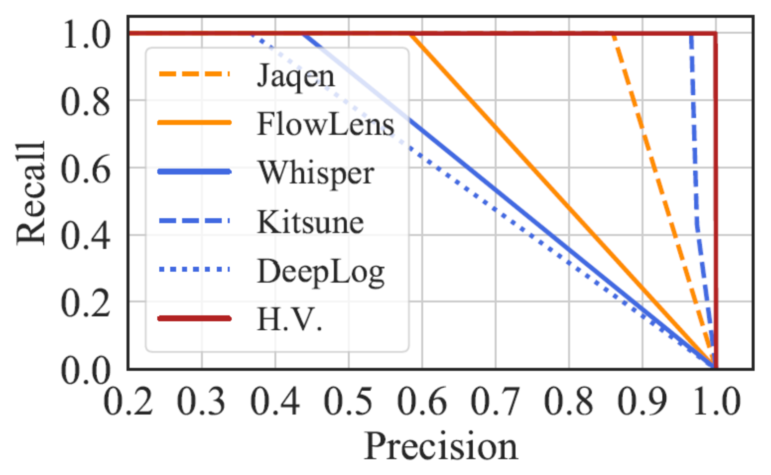
5. Experimental Analysis: Results

- HyperVision outperforms 5 SOTA methods in detection accuracy.
  Over 50% of the stealthy attacks cannot be identified by all the methods.

ROC Comparison

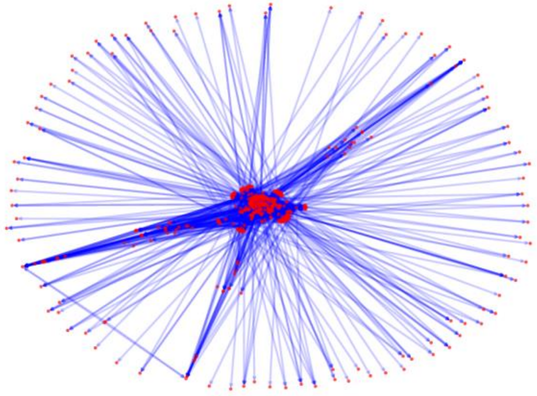


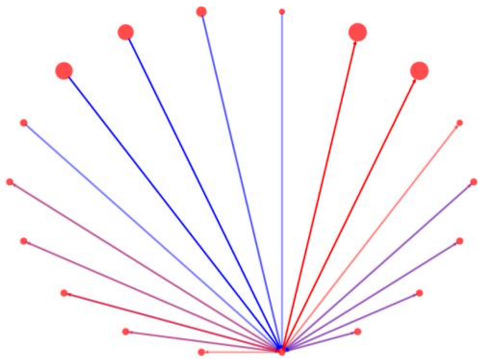
PRC Comparison

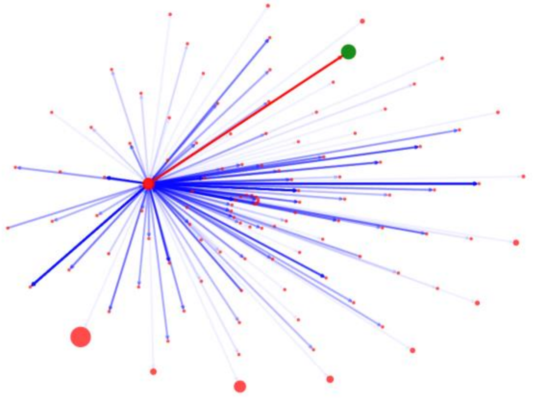


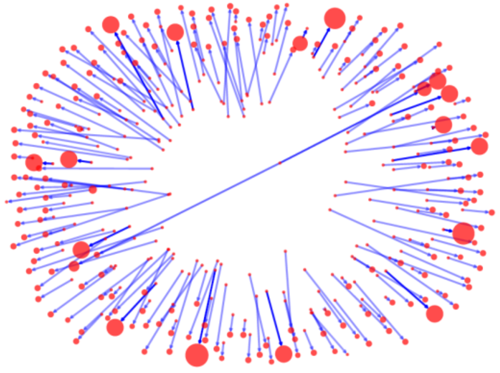
5. Experimental Analysis: results

- The method can detect many sophisticated attacks.


(a) Crossfire.

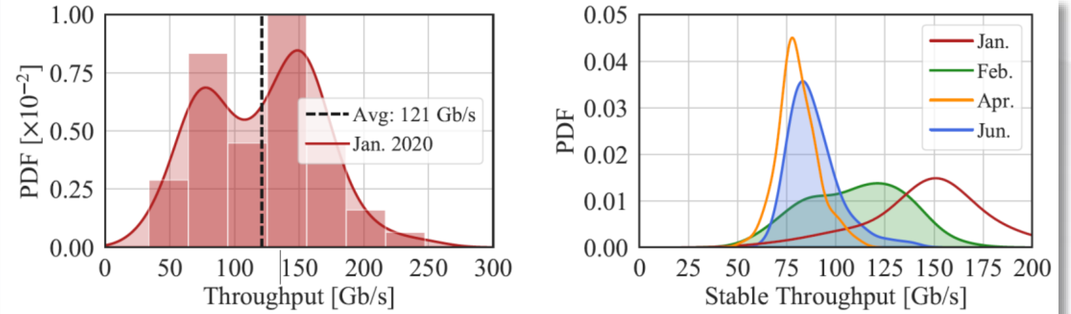

(b) SSH cracking.


(c) XSS detection.

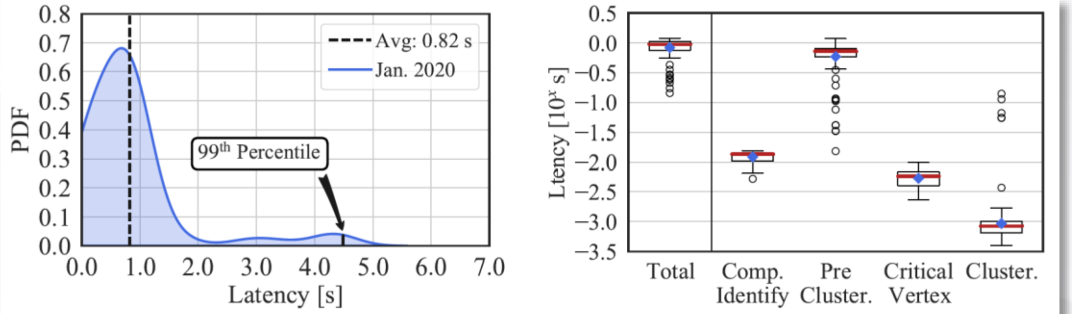

(d) P2P botnet.

5. Experimental Analysis: results

- The method realizes both high detection throughput and low latency.
  - The graph detection module can process 121 Gb/s traffic on average.
  - Meanwhile, the average detection latency is only 0.82s.



(c) Graph detection throughput. (d) Stable detection throughput.

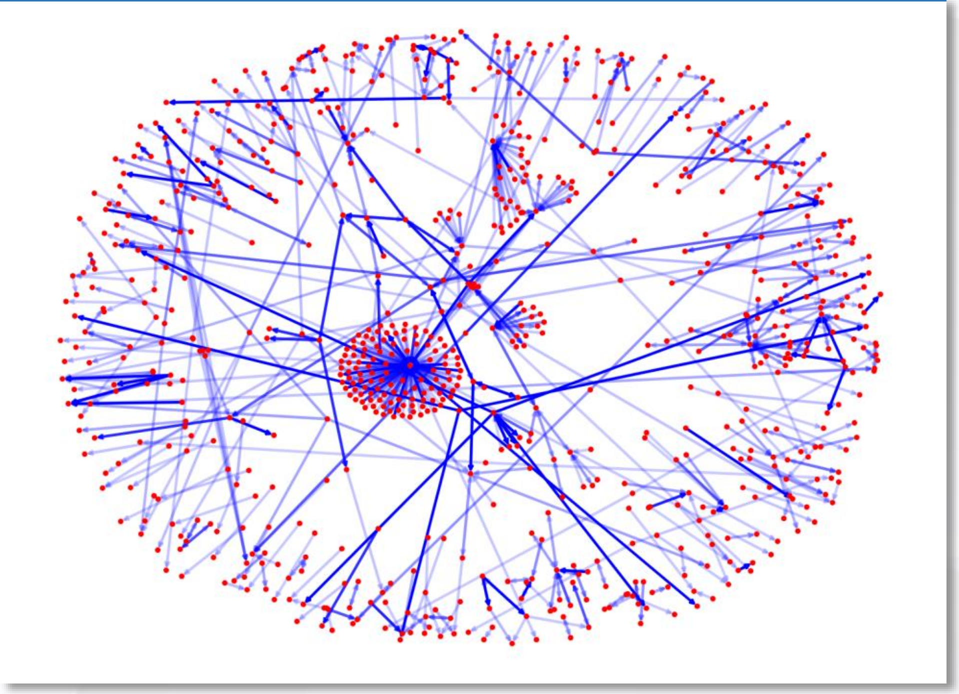


(c) Graph detection latency. (d) Detection latency composition.

6. Conclusions and Takeaways

- We develop an encrypted malicious traffic detection method, which utilize *flow interaction patterns* represented by *graph structural features*.

Flow Interaction Graph



6. Conclusions and Takeaways

- Many attack traffic generates benign traffic features, e.g., packet rates.
- Design new traffic features to tackle this issue.
- The idea of using the graph is derived from provenance graph analysis.

We believe the flow interaction graph can be applied to other network applications.

Detecting Unknown Encrypted Malicious Traffic in Real Time via Flow Interaction Graph Analysis

Effective and Efficient Detection for Encrypted Malicious Traffic

Chuanpu Fu[1], Qi Li[1,2], Ke Xu[1,2]

[1]Tsinghua University, Beijing, China; [2]Zhongguancun Lab, China

NDSS SYMPOSIUM 30TH ANNIVERSARY