DCPsolver: Enhancing DNS Cache Poisoning Defenses with Resolver-Based SmartNICs

Bharath Kollanur
bkolla1@lsu.edu

Kurt Friday
kfriday1@lsu.edu

Elias Bou-Harb
ebouharb@lsu.edu

Louisiana State University, LA, USA

LSU School of Electrical Engineering & Computer Science

Introduction

Problem Statement

- DNS is essential for translating domain names into IP addresses but remains vulnerable to cache-poisoning attacks.
- Traditional defenses mainly address off-path attacks, leaving on-path attacks a major risk.
- Adoption challenges for existing solutions like DNSSEC hinder broader protection.

Objective:

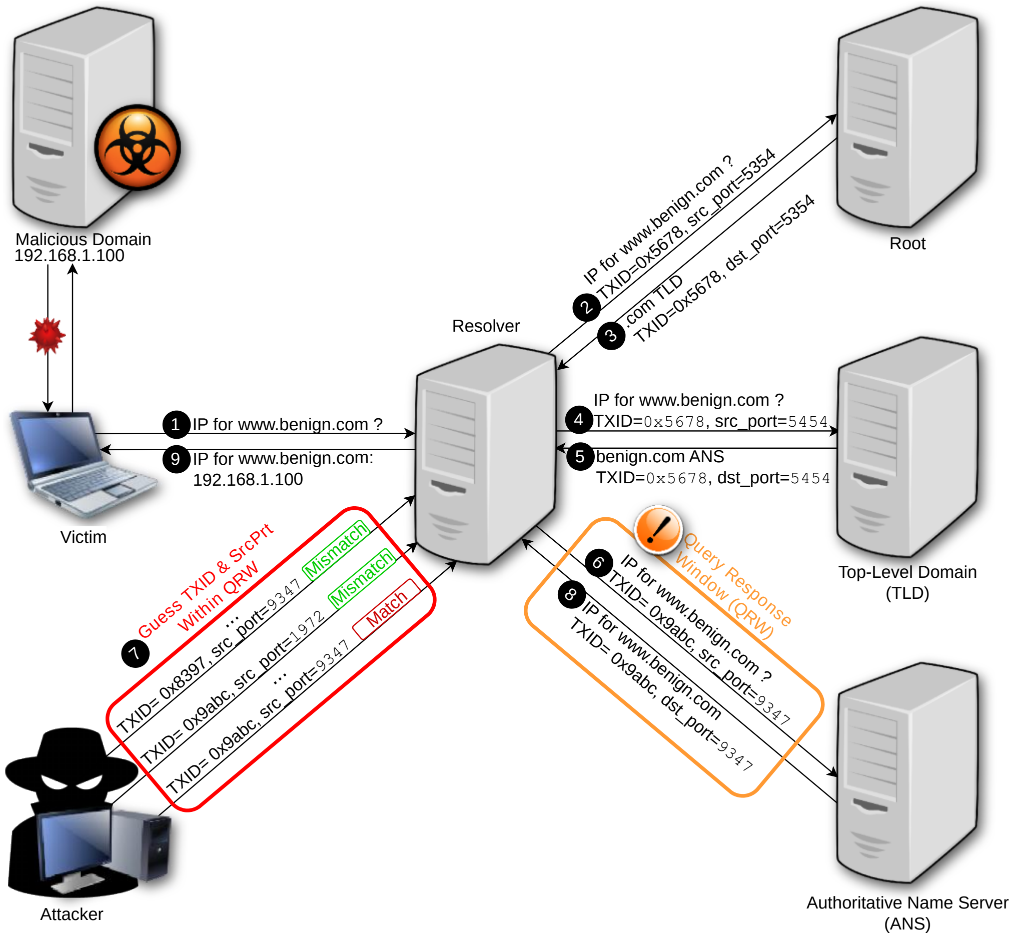
Introduce DCPsolver, a solution leveraging SmartNICs to detect and mitigate on- and off-path DNS Cache Poisoning (DCP) attacks in real-time.

Background

What is DNS Cache Poisoning?

DNS Cache Poisoning is an attack where an adversary injects false DNS records into a resolver's cache, redirecting users to malicious sites.

- **Types of Attacks:**
  - **Off-path attacks:** The attacker guesses the Transaction ID (TXID) and source port, sending spoofed responses to poison the cache.
  - **On-path attacks (Man-in-the-Middle, MITM):** The attacker intercepts DNS communication between the resolver and the authoritative server, injecting false records.



Background

SmartNICs (Smart Network Interface Cards): Programmable NICs that go beyond standard packet forwarding, enabling in-network processing and custom packet manipulation.

Why Use SmartNICs for DNS Security?

- SmartNICs like Nvidia's Bluefield-2 can monitor DNS traffic at speeds up to 100 Gbps, detecting and mitigating attacks in real-time.
- They provide a secure, efficient defense, preventing malicious traffic from reaching the host CPU.
- Unlike DNSSEC, which requires significant infrastructure changes, SmartNICs integrate seamlessly into existing networks.

Threat Model

Assumptions:

- **Attack Surface:** The target system is a recursive DNS resolver vulnerable to both on-path (Man-in-the-Middle, MITM) and off-path attacks.
- **No DNSSEC:** DNSSEC is not implemented, making the resolver susceptible to DNS cache poisoning attacks.

Attack Scenarios:

- The adversary can send DNS queries to the resolver and may intercept legitimate DNS responses.
- The adversary has network access to manipulate packets at key points, particularly during on-path attacks.

Overview of DCPsolver

- **DCPsolver** uses **SmartNICs** for real-time **DNS traffic inspection** and **attack mitigation** directly within the network interface card, reducing CPU load.
- Unlike **DNSSEC**, which demands infrastructure changes and cryptographic overhead, DCPsolver operates entirely within the recursive DNS resolver.
- DCPsolver can detect both **off-path attacks** (e.g., Kaminsky-style) and **on-path attacks** (Man-in-the-Middle).

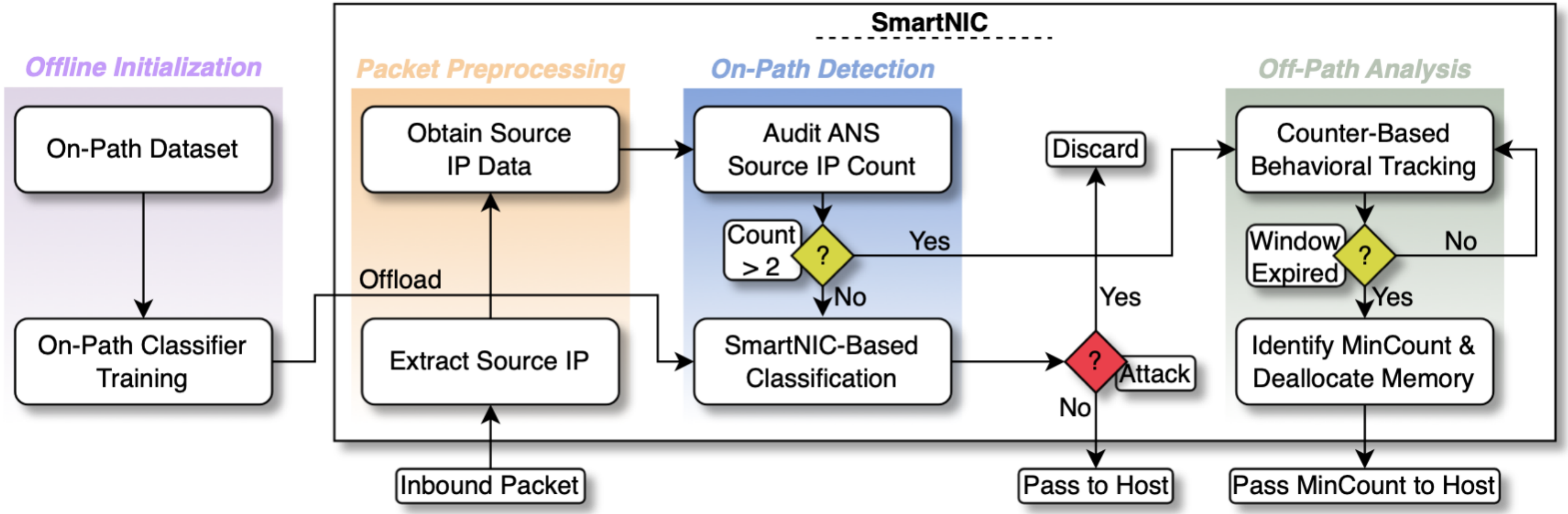
Components of DCPsolver:

- Offline Initialization.
- Packet preprocessing.
- On-path attack detection.
- Off-path analysis.

Offline Initialization

- **Objective:** DCPsolver uses a **Gaussian Naïve Bayes (GaussianNB) classifier** to detect **on-path attacks** by analysing network jitter and Round-Trip Time (RTT).
- **Training Dataset:** The classifier is trained offline using data containing both **benign DNS responses** and **on-path attack traffic.**
- **Features for training:**
  - **Network jitter**
  - **RTT**
- **Training Process:**
  - **GaussianNB** estimates the mean and variance for each feature (jitter, RTT).
  - The classifier is trained to detect deviations in these features, distinguishing between benign and attack traffic.

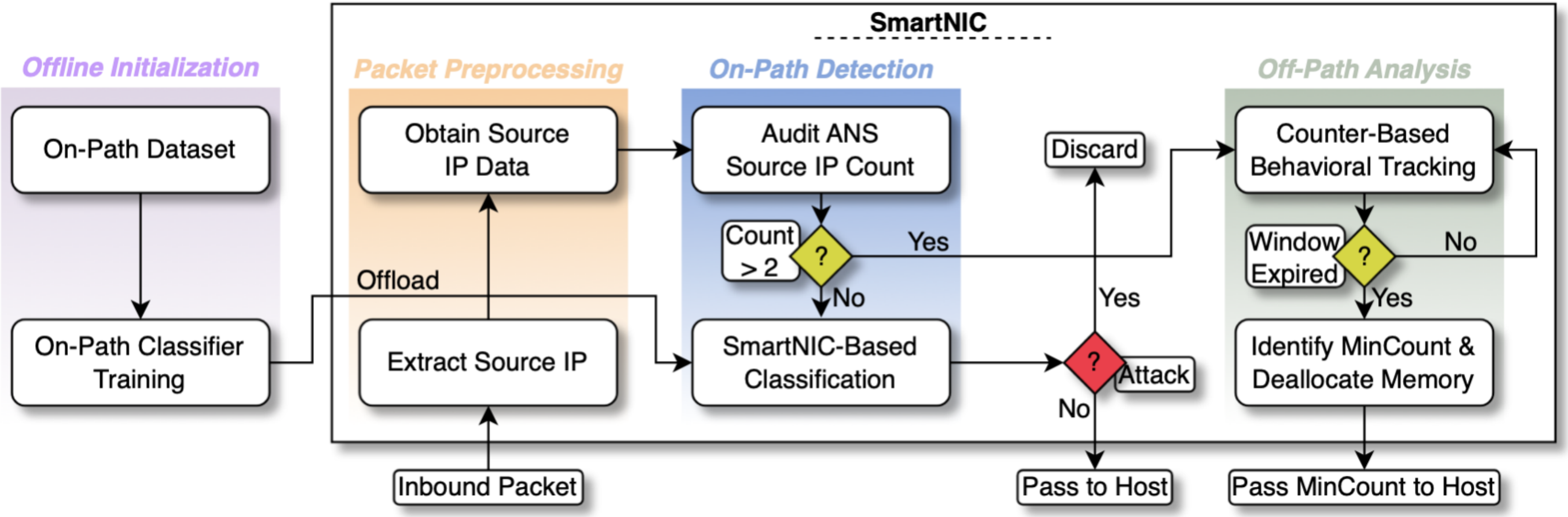
DCPsolver


SmartNIC
Offline Initialization
On-Path Dataset
On-Path Classifier Training
Packet Preprocessing
Obtain Source IP Data
Offload
Extract Source IP
Inbound Packet
On-Path Detection
Audit ANS Source IP Count
Count > 2
?
Yes
No
SmartNIC-Based Classification
Discard
Yes
?
Attack
No
Pass to Host
Off-Path Analysis
Counter-Based Behavioral Tracking
Window Expired
?
No
Yes
Identify MinCount & Deallocate Memory
Pass MinCount to Host

DCPsolver

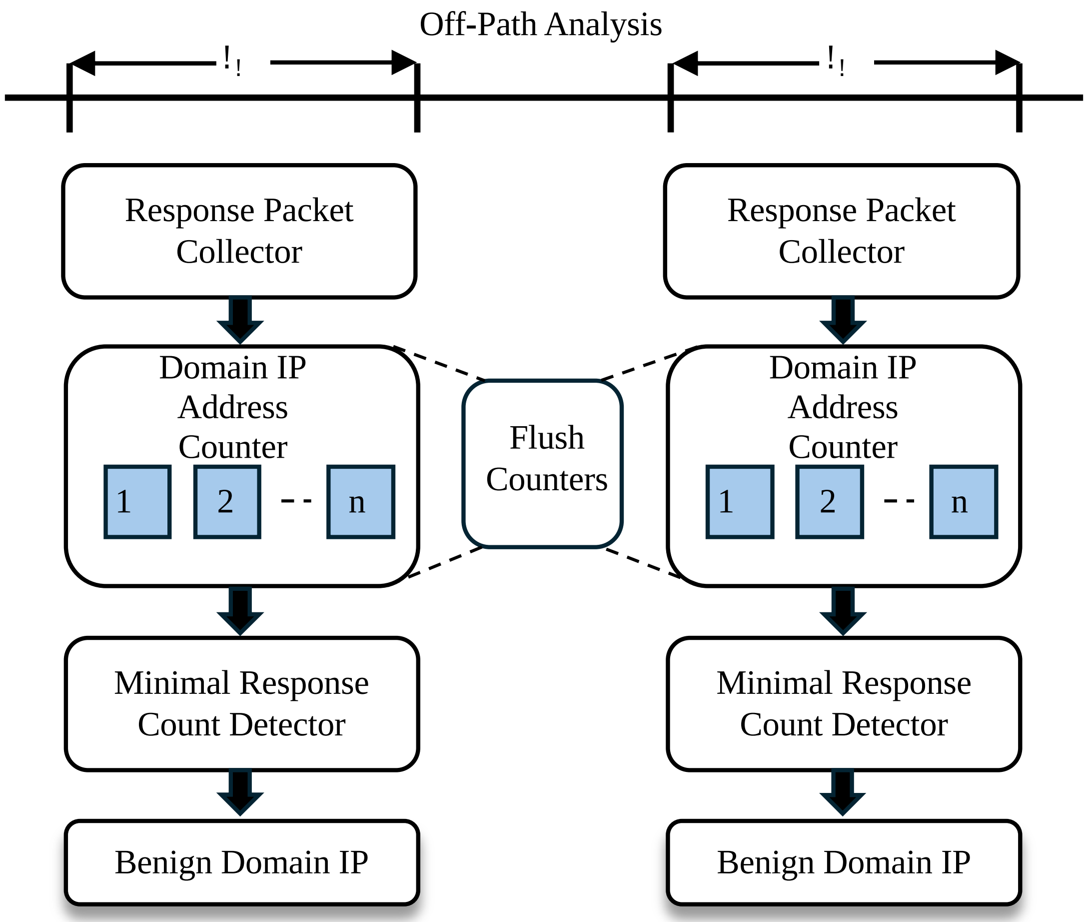
- **SmartNIC Packet Inspection:**
  - As DNS responses arrive, **DCPsolver** extracts key headers like **source IP, transaction ID (TXID), TTL**, and **RTT**.
  - Anomalies are detected in real-time within the SmartNIC, avoiding the need for the traffic to reach the DNS resolver's main CPU.
- **On-path Traffic behaviour:**
  - The system expects two responses: one from the attacker and one from the Authoritative Name Server(ANS).
  - If the packet count is ≤ 2 and the response is classified as benign, the packet is forwarded to the resolver. Otherwise, it is discarded.

DCPsolver


SmartNIC
Offline Initialization
Packet Preprocessing
On-Path Detection
Off-Path Analysis
On-Path Dataset
On-Path Classifier Training
Obtain Source IP Data
Extract Source IP
Offload
Audit ANS Source IP Count
Count > 2
?
Yes
No
SmartNIC-Based Classification
Discard
Yes
?
Attack
No
Counter-Based Behavioral Tracking
Window Expired
?
No
Yes
Identify MinCount & Deallocate Memory
Inbound Packet
Pass to Host
Pass MinCount to Host

Off-path analysis

- **Bucket-Based Counting Mechanism:**
  - Incoming responses are grouped into predefined **time windows (t_w).**
  - The responses for each domain IP are counted during **(t_w).**
  - The source IP with the **minimal count** is classified as benign.



Dataset Generation of DCPsolver Evaluation

- **Lack of pubic dataset:**
  - No public datasets exist for DNS Cache Poisoning (DCP) on-path attacks, so we generated on-path DCP datasets to evaluate **DCPsolver.**
- **Benign data generation:**
  - **Baseline:** CAIDA's IPv4 Routed /24 DNS Names dataset established a baseline of benign DNS activity.
  - **Method:** The CAIDA dataset was replayed through an emulated DNS environment with a Bluefield-2 SmartNIC, providing realistic query and response timings for normal DNS behaviour.

Attack Data Generation

- **Off-path Attack (Kaminsky Style):**
  - A public Kaminsky-style attack tool simulated an off-path attack by flooding the DNS resolver with spoofed responses, attempting to guess the TXID and source port.
- **On-path Attack (MITM):**
  - A Man-in-the-Middle (MITM) attack tool simulated on-path attacks by intercepting DNS queries and injecting malicious responses before legitimate ones.
- **Simulating Real-world Jitter and RTT:**
  - To capture realistic network jitter and RTTs, we used Markov Chain Monte Carlo (MCMC) sampling.

Attack Data Generation

- **Modelling Network Jitter:**
  - Jitter was modelled using **Laplacian distribution.**

$$f(x|\mu, b) = \frac{1}{2b} exp(-\frac{|x-\mu|}{b})$$

  - μ set to the mean jitter values from the **CAIDA dataset** to ensure that the generated jitter data reflects real-world behaviour.
  - b Modelled as a **Half-Normal distribution** to account for variability and outliers typically observed in network jitter data.

Attack Data Generation

- **Modelling RTTs:**
  - RTTs were modelled using a **Gaussian Process (GP)** with an **Exponentiated Quadratic (ExpQuad) covariance function**.
  - The **length scale** of the ExpQuad was drawn from a **Gamma distribution**, capturing the smoothness and variability of real-world RTTs
- **MCMC Sampling:**
  - **Markov Chain Monte Carlo (MCMC)** sampling was used to generate the dataset, specifically employing the **No-U-Turn Sampler (NUTS)**, a variant of Hamiltonian Monte Carlo (HMC), to explore the posterior distribution of jitter and RTT values efficiently.

Evaluation Setup

- **Benign Traffic:**
  - The **CAIDA IPv4 Routed / 24 DNS Names dataset** was replayed through the testbed using tcpreplay to simulate normal DNS traffic.
- **Attack Traffic:**
  - **On-path Attacks:** Man-in-the-Middle (MITM) attacks were simulated using a public attack tool.
  - **Off-path Attacks:** Kaminsky-style cache poisoning attacks were generated by flooding the resolver with spoofed responses, attempting to guess TXID and source port.

Metrics for Evaluation

- **Accuracy:**
  - **Precision:** Proportion of true positive attack detections among all flagged responses.
  - **Recall:** Proportion of true attacks detected out of all attack attempts.
- **Resource Efficiency:**
  - **CPU Utilization:** The system's CPU (SmartNICs) usage during DNS traffic handling and classification.
  - **Off-path Attacks:** Memory consumed by the SmartNIC for packet preprocessing and attack detection.

DCP Attack Detection Results

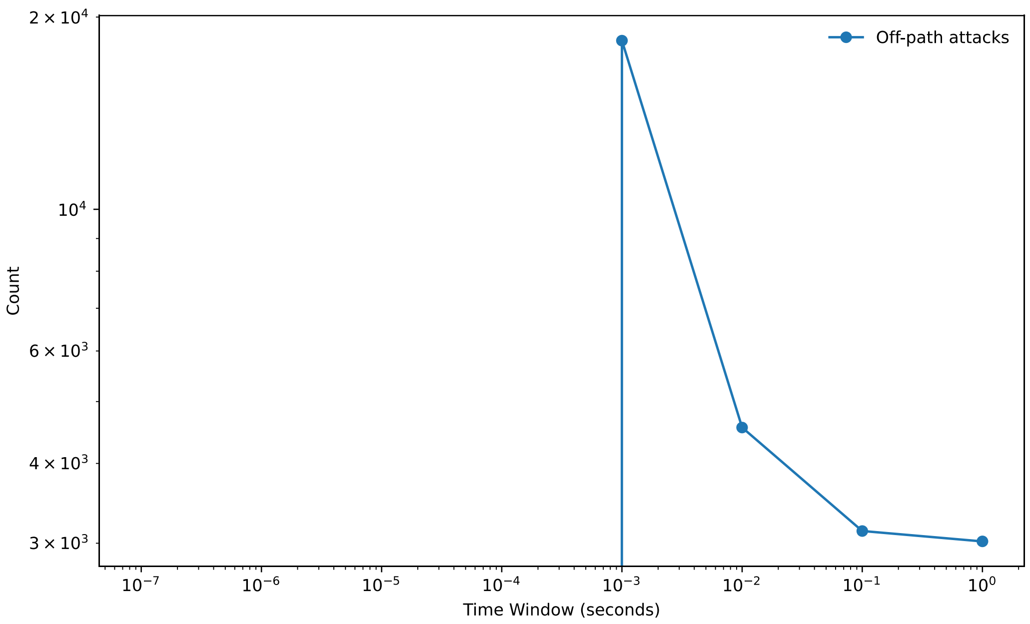
- **Accuracy :**

Class	Precision	Recall	F1-Score
0 (Benign)	0.92	0.98	0.95
1 (Malicious)	0.98	0.92	0.95
Accuracy			0.95

- The **Gaussian Naïve Bayes classifier** accurately flagged attack packets by analysing the jitter values, which followed a **Laplace distribution.**
- The bucket-based counting method successfully flagged off-path attacks by monitoring response counts per domain name.
- Detection accuracy reached **99%** by integrating **TTL** values as thresholds for flagging suspicious responses.

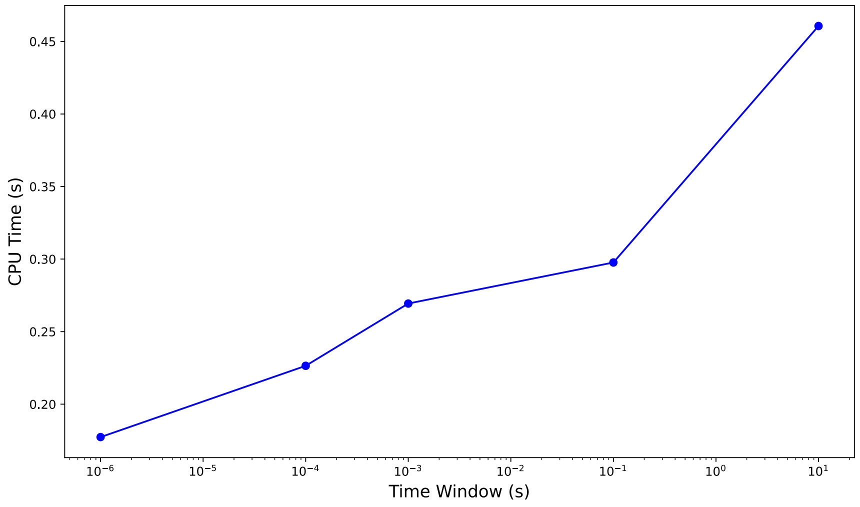
Time Window (t_w) Analysis

- **Detection Sensitivity based on t_w :**
  - **Small Jitter (microsecond range):** Lower detection rate, as fewer responses are clustered.
  - **Medium jitter (millisecond range):** Optimal detection clustering sufficient responses to identify spoofed traffic effectively.
  - **Large jitter:** Higher memory usage with minimal detection improvements, leading to inefficiency.

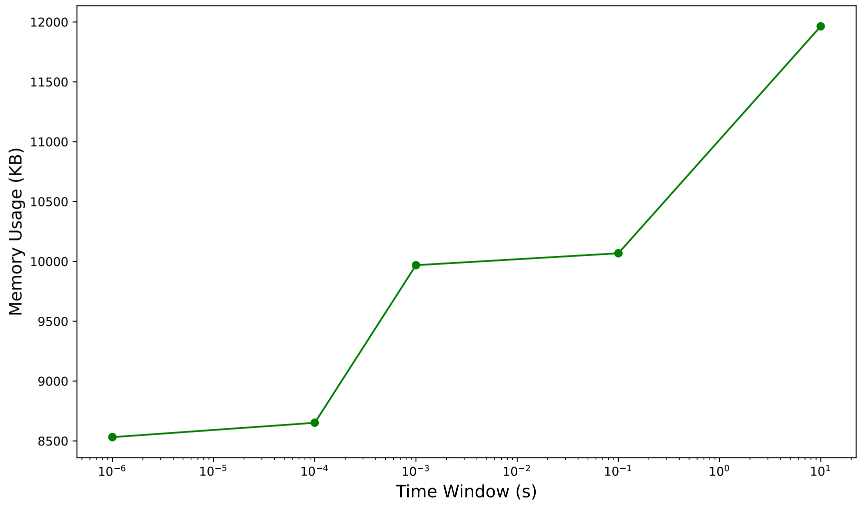


Resource Efficiency

CPU Utilisation



Memory Utilisation



- The ideal t_w for balancing CPU/memory usage and detection accuracy was found to be 10^{-3} seconds.

Conclusion

- DCPsolver provides a practical and scalable solution for DNS cache poisoning defenses both off and on-path, leveraging modern SmartNIC technology.
- Demonstrates **high accuracy** in real-time attack detection while maintaining efficiency.
- **Future Work:**
  - Plan to expand on capturing real-world traffic to enhance dataset accuracy and explore additional attack vectors.

College of Engineering
School of Electrical Engineering & Computer Science

Thank you

Presenter

Bharath Kollanur
bkolla1@lsu.edu

SCAN ME