Towards Accurate and Efficient Detection of Slow Attacks in a Programmable Data Plane

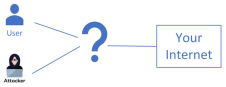
Cuidi Wei, Prof. Timothy Wood (Advisor)
The George Washington University Computer Science

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Abstract

A programmable data plane composed of P4 switches, smartNICs, and hosts running software network functions can provide new opportunities for network security. Much work in this area has focused on monitoring high volume traffic such as denial of service attacks or heavy-hitter detection. However, slow attacks that carefully use small amounts of traffic to have a highly negative effect are much more challenging to detect since they typically require fine-grained analysis of all flows. Our work is exploring how a programmable data plane can provide accurate attack detection at nearly line rate while overcoming challenges such as the limited memory space available on network devices.



Example Slow Attacks

Detecting slow attacks can be more complex compared to monitoring traffic patterns for volumetric attacks due to the distinct features exhibited by slow attacks.

TCP forged RST attack

1. Utilize existing connections and send RST to one or both ends.

2. Difficult to determine if an RST packet is benign or suspicious.

3. Can cause connections lost by sending few packets.

Naive Solution: Delay forwarding RST packets and Track each flow states.

Port Scan Attack

1. Send SYN packets with different ports to target server.

2. If attacker sends packets at low speed, it is difficult to detect.

Common Solution: Estimate the number of traffic packets with distinct ports and reveal the attack.

SSH password attempt attack

1. An attacker only sends a few packets to initiate connection and send password.

2. Contents of SSH packets are encrypted, preventing us from verifying the success or failure of the SSH login.

Naive Solution: Inspect SSH connections to verify the login success or failure.

HTTP related slow attacks

1. Send partial headers to keep the connections with the server open.

2. Sending packets at low speed.

Naive Solution:

Request Header Analysis with deep packet inspection.

Problems and Possible Solutions

Attributes	Naive Solution	Sketch Based Solution	Our Goal
Track every flow	✓	×	✓
Reveal attacks within short time	✓	×	✓
Memory efficient / high performance	×	✓	✓

Research Challenge: How can we provide flow-level, stateful analysis to detect slow attacks given the limited memory of programmable switches and the limited bandwidth/processing speed of end hosts?

Solution Intuition:

- Filter traffic to quickly redirect known benign flows
- Split flow table across fast but small memory switch and slower but high memory smartNIC and host
- Batch and merge updates from switch to host to reduce communication costs

Design and Architecture

In general, many slow attacks can be detected by: 1. Tracking the flow states, 2. Monitoring the number of packets, 3. Measuring communication rate.

Our Design include:

- Hierarchical Data plane
  - Fastest Path, Fast Path and Smart Path
- Dynamic Bloom Filter
  - Use Dynamic Bloom Filter to keep active safe flows and clear unactive flows.
- Efficient Flow Tables across P4 Switch and SmartNIC
  - Merge information from programmable switch, smartNIC, and host to detect distributed slow attacks across multiple routes.

Hierarchical Data plane

Fastest Path: for traffic that we have already whitelisted and can forward with no analysis and minimal state tracking (just bloom filter bits)
Fast Path: for traffic that can be analyzed using state already on the P4 switch (small size flow table)
Smart Path: for traffic that must go to the smartNIC for more detailed analysis or access to state not available on the switch (large flow table)

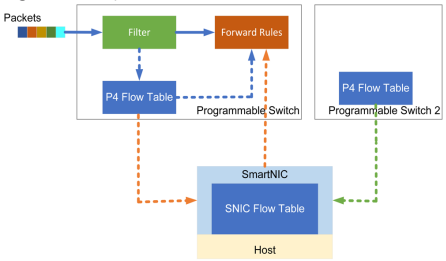
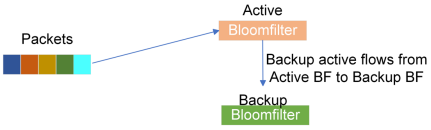


Figure 1. Architecture of Slow Attack Detection Model

Dynamic Bloom Filter

Active BF and Backup BF.
Goals: Clear unactive flows but keep active flows.



1. Backup BF change to Active BF and flush Active BF.
2. Change empty Active BF to Backup BF.

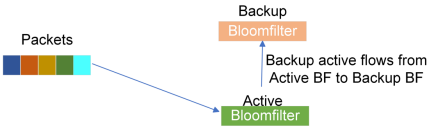


Figure 2. Dynamic Bloom Filter

Efficient Flow Tables

1. State Transitions of Flow Table

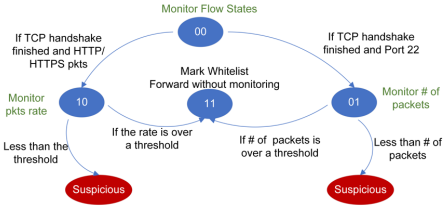


Figure 3. State Transitions of Flow Table

Design and Architecture

2. Merging traffic information between multiple switches and SmartNIC

- P4 Flow Table sends its flow table with a batch of flow entries to SNIC.
- SNIC merges all flow entries to SNIC Flow Table.

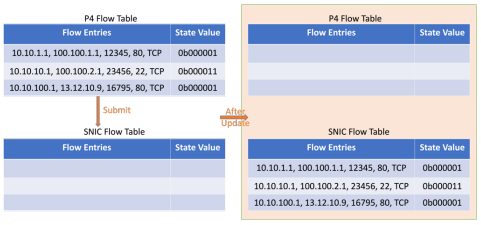


Figure 4. Merging traffic information

- When merging flow entries of P4 Flow Table and SNIC Flow Table, SNIC Flow Table do ADD operation.

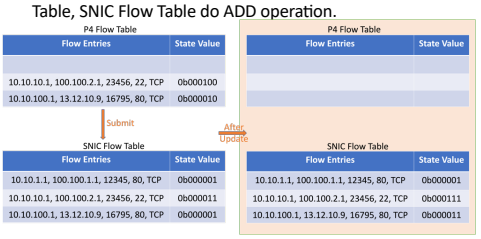


Figure 5. Merging traffic information

Performance Factors

To predict the performance, there are some factors impacting the performance:

- Fraction of analyzed packets of whole traffic packets.

Flow Type	# packets analyzed	The flow packets Ratio of total TCP traffic
TCP handshake	3	100%
SSH connection	~50	Less 1%
HTTP	A few	25%
HTTPS	A few	50%

Table 2. Fraction of analyzed type of TCP packets

- Fraction of flows in P4 Flow Table or in SNIC Flow Table
- Communication frequency and latency between P4 switch and SmartNIC.

References

[1] Xiaoqi Chen, Shir Landau-Feibish, Mark Braverman, and Jennifer Rexford. 2020. Beaucoup: Answering many network traffic queries, one memory update at a time. In Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. 226–239.

[2] Sourav Panda, Yixiao Feng, Sameer G Kulkarni, KK Ramakrishnan, Nick Duffield, and Laxmi N Bhuyan. 2021. SmartWatch: accurate traffic analysis and flow-state tracking for intrusion prevention using SmartNICs. In Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies. 60–75.

[3] Satadal Sengupta, Hyojoon Kim, and Jennifer Rexford. 2022. Continuous in-network round-trip time monitoring. In Proceedings of the ACM SIGCOMM 2022 Conference. 473–485

[4] Jurkiewicz, P., Rzym, G., & Boryło, P. (2021). Flow length and size distributions in campus Internet traffic. *Computer Communications*, *167*, 15-30.

This work was supported in part by NSF Grant #2210380.