Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.3, June 2026, pp. 2219~2227
e-ISSN: 2548-6861

Performance Analysis of Suricata as an Intrusion Detection System (IDS) in Detecting Slowloris Attacks on Web Servers

Andika Agus Slameto [1]*, Eka Marlina Kemala Sari[2]*
* Informatika, Universitas Amikom Yogyakarta
rmkt.andika@amikom.ac.id [1], ekamarlinakemalasari@students.amikom.ac.id [2]

Article Info

Article history:

Received 2026-04-02
Revised 2026-04-29
Accepted 2026-05-05

Keyword:

Intrusion Detection System,
Network Security,
Slowloris,
Suricata,
Web Server.

ABSTRACT

Network security on web servers is a crucial element for ensuring service availability. Slowloris represents a low-rate variant of the Denial-of-Service (DoS) attack, leveraging HTTP connection handling mechanisms by submitting perpetually incomplete requests, which deplete server connection slots without necessitating high bandwidth. This study evaluated Suricata as an Intrusion Detection System (IDS) on an Apache web server through 50 controlled Slowloris attack simulations within a VirtualBox virtual environment running Ubuntu 24.04 LTS, using a single custom detection rule targeting connection-rate patterns. Three performance parameters were analyzed: (1) Detection speed, defined as the elapsed time from the Slowloris script's first socket-loop execution (T_0) to the first alert in Suricata's fast.log (T_1); (2) Detection rate (True Positive Rate), determined via a confusion matrix; and (3) System resource consumption of the Suricata process (CPU, RAM, and bandwidth), benchmarked against a no-IDS reference baseline. Results indicated an average detection time of 0.346 seconds (min 0.168 s, max 0.979 s), a mean detection rate of 72.84% (min 50.00%, max 94.12%), a mean IDS-attributable CPU overhead of approximately 5.47 percentage points above the no-IDS server baseline, a 6.39 MB increase in RAM (< 0.5% of allocated VM memory), and a 256.72 kbps increase in bandwidth. No false positives were observed across all 50 attack-only trials; however, this finding was not validated under concurrent normal traffic conditions. Cross-parameter analysis revealed: a statistically significant positive correlation between attack bandwidth and detection rate (Pearson $r = 0.468$, $t(48) = 3.67$, $p < 0.001$); a non-linear relationship between detection time and detection rate (quadratic $R^2 = 0.18$, $p = 0.009$) with an empirically validated optimal window of 0.25–0.35 s associated with detection rates above 90%; and negligible correlations for ΔCPU ($r = 0.072$) and ΔRAM ($r = 0.021$) with detection rate. These findings confirm Suricata's effectiveness as a lightweight early-warning mechanism for Slowloris mitigation on standard-specification web servers, with generalisability to physical environments and other DoS attack types requiring further validation.

I. INTRODUCTION

The expansion of Information technology has increasingly relied on web-based services across education, business, and public administration sectors, positioning web servers as critical infrastructure and prime targets for cyberattacks [1]. Apache HTTP Server, one of the most widely used open-source web servers, dominated global market share according to the Netcraft January 2023 survey [2], making it a representative platform for security research. Among threats targeting service availability, the Slowloris attack is particularly insidious. Unlike volumetric Distributed Denial-of-Service (DDoS) attacks, which exploit HTTP connection-handling mechanisms, attackers open hundreds of TCP connections to a target server and keep them active by periodically sending incomplete HTTP headers, leaving the

request incomplete [3]. This exhausts all available connection slots, preventing legitimate users from accessing the service. Harefa et al. [4] documented up to 50% increases in local network infrastructure congestion and severe response-time degradation under Slowloris attacks, confirming the real-world severity of these threats. Intrusion Detection Systems (IDSs) are a crucial part of a defense-in-depth strategy, monitoring network traffic in real time and generating alerts when they detect attack patterns. Suricata, developed by the Open Information Security Foundation (OISF), stands out among open-source IDS solutions due to its multi-threaded architecture, Deep Packet Inspection (DPI) capabilities, and flexible rule-signature engine [5]. Prior studies have demonstrated Suricata's competitive performance across various scenarios. Ralianto and Cahyono [6] reported a mean detection accuracy of 61%, significantly outperforming Snort (31%) in Pytbull-based tests. Bada et al. [7] documented Suricata's detection rate (TPR) of 97.40% over an 18-hour evaluation period, covering DoS, probe, and scanning attacks. Lukman and Suci [8] found that Suricata was more effective than Snort in SYN Flood scenarios. Simarmata et al. [9] integrated Random Forest into Suricata for Slow Read detection—a low-rate attack variant similar to Slowloris—achieving 94% accuracy. Stephani et al. [10] deployed Suricata on OPNsense and successfully detected Slowloris, along with other DDoS threats.

While several prior studies have evaluated IDS performance against DoS attacks [6][7][8][9][10], three specific gaps remain unaddressed. First, no study has simultaneously measured detection speed, confusion-matrix-based accuracy, and all three system resource metrics (CPU, RAM, bandwidth) for Suricata specifically against Slowloris. Second, no prior work has quantitatively analyzed cross-parameter relationships—particularly the nonlinear interactions among detection time, bandwidth intensity, and detection accuracy—in a Slowloris context. Third, existing Slowloris evaluations (e.g., Stephani et al. [10]) have not reported per-trial breakdown statistics nor identified an empirically validated optimal detection-time window. This study addresses these gaps through 50 controlled simulations, with three primary contributions: (1) the empirical identification of an optimal detection-time range of 0.25–0.35 s that maximizes accuracy; (2) statistical quantification of the bandwidth–detection rate relationship (Pearson $r = 0.468$, $p < 0.001$); and (3) a simultaneous three-resource profile (CPU, RAM, bandwidth) of Suricata under controlled Slowloris conditions. These contributions are evaluated within a VirtualBox virtual environment, which—while enabling reproducible controlled conditions—imposes known constraints relative to physical network deployments, as discussed in Section III.

II. METHOD

This study employs a quantitative experimental design with a virtual simulation environment. The independent variable is the Slowloris attack simulation (50 trials); dependent variables include detection speed (s), detection accuracy (%), and Suricata process resource consumption: CPU (%), RAM (MB), and bandwidth (kbps). Table 1 details the test environment specifications.

TABLE 1
TEST ENVIRONMENT SPECIFICATION

Component	Specification
Host Processor	Intel Core i5-8350U
Host Memory	16 GB
Virtualization	Oracle VirtualBox
VM Server (RAM/CPU)	5,228 MB / 4 Cores
VM Attacker (RAM/CPU)	4,096 MB / 4 Cores
VM Operating System	Ubuntu 24.04.1 LTS
Web Server	Apache2
IDS	Suricata 8.0.2
Attack Tool	Slowloris (Python, modified)
Network Segment	192.168.43.0/24 (Bridged)

The network was configured using Bridged Adapter mode (virtio-net), positioning the VM Server (192.168.43.124) and VM Attacker (192.168.43.97) on the same network segment. Promiscuous Mode was configured to monitor only packets destined for the server. The Slowloris script was adjusted to record timestamps with millisecond precision, facilitating precise measurement of the elapsed time between attack execution and alert appearance in fast.log. The host machine ran an Intel Core i5-8350U processor with 16 GB physical RAM; the VM Server was allocated 5,228 MB RAM across 4 virtual CPU cores, and the VM Attacker was allocated 4,096 MB RAM across 4 virtual CPU cores. Both VMs ran Ubuntu 24.04.1 LTS. Suricata 8.0.2 was installed from the OISF official repository, with af-packet configured as the capture method on interface enp0s3. The default Suricata threading model (auto-detect) was used, with flow and packet threads automatically assigned based on available CPU cores. HOME_NET was set to 192.168.43.0/24 in suricata.yaml. No additional ruleset (e.g., Emerging Threats) was loaded; only the single custom rule described below was active, to isolate Slowloris-specific detection behavior.

The Suricata detection rule was configured to detect Slowloris connection patterns on port 80: `alert tcp any any -> $HOME_NET 80 (msg: "Potential Slowloris Attack"; flow:to_server, established; threshold: type both, track by_src, count 100, seconds 1; classtype:attempted-dos; sid:1000001; rev:4;)`. A threshold of 100 connections per second was selected because it far exceeds the typical tens to low hundreds of connections per second from a single legitimate HTTP client. The `flow:to_server, established` parameter restricts analysis to active inbound connections, thereby reducing processing overhead. HOME_NET was defined as 192.168.43.0/24 in Suricata.yaml, with af-packet capture configured on interface enp0s3.

Each trial consisted of two phases: (1) a baseline phase (~2 minutes) for recording normal CPU, RAM, and bandwidth via htop and iftop; and (2) an attack phase (~5 minutes) executing Slowloris with 5000 sockets. VM constraints limited the

number of simultaneous connections to approximately 600, yet this was sufficient to trigger the detection rule. The cycle was repeated 50 times. Although the socket target was fixed at 5,000 across all trials, natural variation in VM scheduler behavior, OS TCP stack state, and CPU load produced per-trial differences in the number of bursts (14–18), effective connection count (~400–650), and inter-burst intervals. This emergent variation yielded a range of attack bandwidth values (124–802 kbps) and detection times (0.168–0.979 s per trial, mean), providing sufficient cross-trial variance for correlation analysis without requiring manual manipulation of attack parameters. Data analysis used descriptive statistics [16], with accuracy computed as:

$$\text{Accuracy} = (TP+TN) / (TP + TN+ FP+ FN) \quad (1)$$

The term "burst" refers to each wave of connections sent by Slowloris within a single trial interval. In this study, TN = FP = 0 because the evaluation was conducted exclusively under active attack conditions; consequently, Equation (1) reduces to TP / (TP + FN), which is equivalent to Recall (True Positive Rate). This metric is referred to as the detection rate throughout this paper to avoid terminological ambiguity with standard accuracy metrics that require all four confusion-matrix components.

III. Result And Discussion

A. *Detection Speed*

Detection time was measured from the timestamp of Slowloris execution on the attacker's terminal to the first alert in Suricata's fast.log for each trial. Operationally, the attack start time (T_0) was defined as the Unix timestamp (millisecond precision) recorded by the modified Slowloris script immediately before the first socket-opening loop began. The alert time (T_1) was defined as the timestamp printed in fast.log by Suricata for the first matched alert. Detection time per burst was then calculated as $T_1 - T_0$ for each detected burst. Both machines used NTP-synchronized clocks; clock drift between VMs was verified to be below 5 ms across all sessions. The per-trial mean detection time reported in Table 2 is the arithmetic mean of all individual burst detection times within that trial. Table 2 presents the complete dataset for all 50 trials, and Figure 1 visualizes their distribution.

TABLE 2.
SURICATA DETECTION SPEED — 50 TRIALS

No	Bursts	Det.	Mean (s)	SD	Min	Max
1	15	9	0.196	0.246	0.015	0.874
2	17	13	0.302	0.175	0.014	0.549
3	15	10	0.442	0.290	0.088	0.987
4	17	12	0.277	0.132	0.081	0.585
5	15	10	0.582	0.852	0.003	3.015
6	17	13	0.281	0.167	0.037	0.568
7	15	9	0.384	0.271	0.082	0.865
8	15	10	0.225	0.216	0.018	0.699
9	15	10	0.212	0.201	0.007	0.637
10	17	12	0.356	0.130	0.253	0.589
11	18	12	0.359	0.234	0.088	0.834
12	15	8	0.431	0.206	0.125	0.847
13	16	13	0.342	0.182	0.025	0.632
14	15	12	0.561	0.486	0.048	1.920
15	15	11	0.402	0.289	0.138	1.088
16	16	11	0.311	0.195	0.001	0.695
17	15	9	0.297	0.243	0.005	0.170
18	17	14	0.303	0.206	0.010	0.616
19	15	14	0.338	0.275	0.010	0.940
20	15	9	0.217	0.235	0.029	0.824
21	17	13	0.333	0.166	0.055	0.760
22	15	9	0.335	0.054	0.056	0.868
23	16	13	0.220	0.089	0.061	0.427
24	16	14	0.309	0.255	0.036	0.808
25	17	16	0.314	0.251	0.094	1.067
26	14	7	0.979	1.421	0.174	4.441
27	15	11	0.341	0.531	0.016	1.958
28	17	13	0.339	0.175	0.113	0.584
29	14	9	0.509	0.491	0.068	1.797
30	17	15	0.168	0.096	0.032	0.330
31	14	10	0.521	0.517	0.054	1.856
32	17	15	0.248	0.192	0.029	0.813
33	15	12	0.413	0.442	0.094	1.812
34	15	8	0.268	0.299	0.013	0.997
35	15	11	0.419	0.303	0.011	0.906
36	17	12	0.266	0.175	0.043	0.640
37	15	10	0.292	0.238	0.041	0.841
38	17	11	0.273	0.147	0.045	0.587
39	17	14	0.302	0.250	0.038	1.063
40	17	14	0.228	0.114	0.006	0.549
41	15	11	0.440	0.201	0.103	0.797
42	15	9	0.662	0.447	0.222	1.795
43	17	12	0.199	0.094	0.039	0.376
44	17	16	0.250	0.162	0.035	0.603
45	15	11	0.426	0.229	0.102	0.787
46	15	11	0.333	0.287	0.055	0.894
47	16	13	0.272	0.247	0.012	0.735
48	15	9	0.236	0.227	0.008	0.665
49	15	13	0.331	0.252	0.022	0.934
50	17	13	0.254	0.204	0.054	0.791

Det. = detected bursts; Mean = mean detection time (s); SD = standard deviation; Min/Max = fastest/slowest detection per trial.

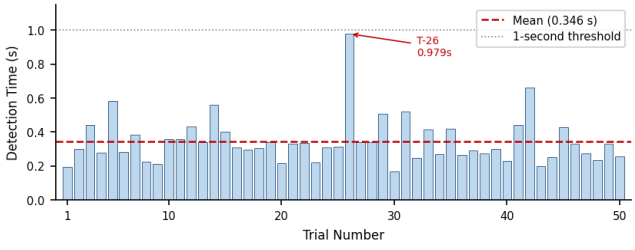

Figure 1. Mean Detection Time per Trial

The mean detection time across all 50 trials was 0.346 seconds. All per-trial mean values remained below 1 second, with the fastest recorded at 0.168 seconds (Trial 30) and the slowest at 0.979 seconds (Trial 26). The highest within-trial

standard deviation was 1.421 seconds (Trial 26), with an intra-trial maximum of 4.441 seconds, suggesting intermittent instability in that session. The overall mean standard deviation across trials was 0.276 s, indicating that per-trial fluctuations were generally contained. These findings are consistent with Zain et al. [11], who reported sub-second detection times with optimized rules, and corroborate the responsiveness results documented by Lukman and Suci [8] and Simarmata et al. [9].

B. Detection Accuracy

Per-trial detection rate was computed using Equation (1). Because the evaluation was conducted exclusively under active attack conditions, TN and FP were both zero; consequently, the formula reduces to TP/(TP+FN), which is equivalent to Recall (True Positive Rate). This metric is hereafter referred to as the detection rate to avoid terminological ambiguity. Table 3 reports TP, FN, and the resulting detection rate for all 50 trials; Figure 2 visualizes the distribution.

TABLE 3.
SURICATA DETECTION ACCURACY — 50 TRIALS

No	Bursts	TP	FN	Accuracy (%)
1	15	9	6	60.0
2	17	13	4	76.5
3	15	10	5	66.7
4	17	12	5	70.6
5	15	10	5	66.7
6	17	13	4	76.5
7	15	9	6	60.0
8	15	10	5	66.7
9	15	10	5	66.7
10	17	12	5	70.6
11	16	12	4	75.0
12	15	8	7	53.3
13	16	13	3	81.3
14	15	12	3	80.0
15	15	11	4	73.3
16	16	11	5	68.8
17	15	9	6	60.0
18	17	14	3	82.4
19	15	14	1	93.3
20	15	9	6	60.0
21	17	13	4	76.5
22	15	9	6	60.0
23	16	13	3	81.3
24	16	14	2	87.5
25	17	16	1	94.1
26	14	7	7	50.0
27	15	11	4	73.3
28	17	13	4	76.5
29	14	9	5	64.3
30	17	15	2	88.2
31	14	10	4	71.4
32	17	15	2	88.2
33	15	12	3	80.0
34	15	8	7	53.3
35	15	11	4	73.3
36	17	12	5	70.6
37	15	10	5	66.7
38	17	11	6	64.7
39	17	14	3	82.4
40	17	14	3	82.4
41	15	11	4	73.3
42	15	9	6	60.0
43	17	12	5	70.6
44	17	16	1	94.1
45	15	11	4	73.3
46	15	11	4	73.3
47	16	13	3	81.3
48	15	9	6	60.0
49	15	13	2	86.7
50	17	13	4	76.5

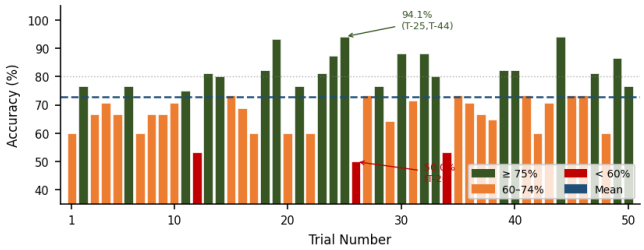

Figure 2. Detection Accuracy per Trial

The mean detection rate was 72.84%, ranging from 50.00% to 94.12%. The peak detection rate was 94.12%, recorded in Trials 25 and 44; the lowest was 50.00% in Trial 26. Of the 50 trials, 15 (30%) achieved a detection rate ≥80%, and 3 (6%) exceeded 90%. While these figures confirm Suricata's capability as an early-warning system, the moderate mean of 72.84% also warrants critical examination. The rate below 100% is a structural consequence of the count-based threshold rule: Slowloris operates at low connection rates by design, and burst waves that do not reach 100 connections within a 1-second window are invisible to the rule as configured. Specifically, False negatives (mean 4.24 bursts/trial) arose from two factors: (1) Slowloris's low-rate nature, whereby some burst waves did not satisfy the count 100/s threshold; and (2) HTTP keep-alive headers that mimic normal traffic patterns. This suggests the 72.84% figure reflects a rule-sensitivity limitation rather than a Suricata architectural weakness, as Simarmata et al. [9] demonstrated that augmenting Suricata with a machine-learning classifier (Random Forest) on Slow Read attacks achieved 94% accuracy—a finding that supports the hypothesis that algorithmic detection layers can compensate for threshold-based blind spots. Bada et al. [7] reported a 70–80% detection rate for layer-7 attacks as competitive performance, contextualizing the 72.84% result as within the acceptable range for rule-based IDS against low-rate DoS. Stephani et al. [10] also confirmed Suricata's effectiveness as an early-warning layer even at sub-100% detection rates. Regarding false positives: FP = 0 was consistently observed across all 50 trials. However, it is important to note that all trials were conducted under active attack conditions with no concurrent

normal (background) traffic. While this controlled setting confirms the rule's high precision in isolating attack bursts from baseline noise—as the 2-minute pre-attack baseline contained zero alerts—it does not fully validate precision under mixed-traffic conditions. The impact of legitimate high-concurrency HTTP traffic (e.g., from web crawlers or CDN prefetchers) on false-positive rates remains untested and represents a limitation acknowledged in Section IV.

C. System Resource Consumption

Table 4 presents CPU, RAM, and bandwidth usage for all 50 trials under both baseline and attack conditions. Table 5 summarizes the descriptive statistics.

TABLE 4.
SYSTEM RESOURCE CONSUMPTION — 50 TRIALS

No	CPU Base (%)	CPU Atk (%)	RAM Base (MB)	RAM Atk (MB)	BW Base	BW Atk (kbps)
1	2.6	4.2	61.0	66.8	3.7	282
2	2.1	36.4	59.3	66.3	3.7	254
3	3.3	5.7	58.7	66.2	3.5	268
4	2.0	2.0	60.1	66.1	1.2	251
5	2.5	3.2	59.0	65.9	3.8	192
6	1.1	2.1	61.0	67.3	3.9	154
7	2.7	4.7	59.7	66.0	3.4	155
8	2.5	8.1	59.0	66.1	2.0	194
9	3.3	4.6	59.1	66.2	1.5	257
10	1.4	3.0	61.0	66.9	1.2	297
11	2.4	7.7	61.3	67.2	2.0	157
12	2.0	6.4	60.2	66.0	1.9	124
13	2.0	8.3	58.9	66.1	1.7	298
14	2.6	3.5	59.2	66.3	2.0	164
15	2.0	5.3	59.2	65.9	2.0	181
16	1.4	2.2	60.3	65.9	1.9	259
17	2.0	5.6	58.9	66.1	2.0	161
18	1.5	8.3	58.9	66.3	2.2	297
19	3.0	4.7	58.9	65.9	2.2	234
20	2.6	20.5	59.9	65.8	2.3	292
21	1.3	7.7	60.0	66.2	1.3	298
22	2.1	7.0	58.9	65.9	1.7	233
23	1.2	1.4	59.2	65.6	1.4	353
24	2.2	16.2	60.9	66.6	2.3	227
25	1.4	2.3	60.2	66.8	1.7	802
26	2.6	4.1	59.6	65.9	2.3	130
27	2.0	6.0	59.0	66.0	2.1	200
28	1.6	2.1	61.5	67.5	1.6	298
29	2.2	3.7	60.0	65.9	2.3	149
30	1.4	2.3	59.6	66.0	1.5	298
31	2.0	4.4	60.1	66.3	1.9	258
32	1.3	13.3	60.2	66.1	1.5	328
33	3.1	5.3	59.1	66.0	1.7	235
34	2.1	5.2	58.7	65.7	1.1	275
35	2.2	4.3	61.0	66.9	2.6	249
36	1.3	12.5	58.9	66.3	1.5	249
37	2.2	5.9	59.0	65.9	1.7	202
38	1.8	2.7	59.2	66.0	2.0	296
39	1.4	11.8	59.4	66.0	1.7	313
40	1.3	2.0	59.6	66.0	1.7	323
41	2.1	7.2	59.0	66.2	1.4	175
42	2.6	6.9	61.5	66.9	1.8	283
43	1.3	1.6	63.5	66.8	1.4	297
44	1.4	1.4	60.7	66.5	1.9	297
45	2.4	23.5	59.9	65.6	1.0	254
46	1.9	5.1	58.8	65.9	2.0	257
47	2.0	29.3	60.6	66.6	2.0	296
48	2.0	13.0	60.4	66.1	2.5	224
49	2.7	6.4	61.0	66.6	2.3	256
50	1.3	1.4	59.6	65.9	1.9	310

Base = baseline (pre-attack); Atk = under attack; BW = bandwidth.

TABLE 5.
DESCRIPTIVE STATISTICS OF RESOURCE CONSUMPTION (N = 50)

Parameter	Mean Baseline	Mean Under Attack	Δ Mean
CPU (%)	2.03 (max 3.3)	7.25 (max 36.4)	+5.22
RAM (MB)	59.85 (min 58.7)	66.24 (max 67.5)	+6.39
Bandwidth (kbps)	2.04 (max 3.9)	256.72 (max 802)	+254.68

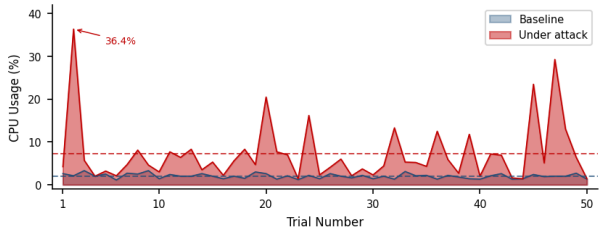

Figure 3. CPU Usage: Baseline vs. Under Attack

Mean CPU usage increased by +5.22 percentage points (from 2.03% baseline to 7.25% under attack; Figure 3). The majority of trials (43/50) recorded increases below 15%, consistent with Ralianto et al. [6] and Bada et al. [7]. Significant outliers were observed in Trials 2 (36.4%), 47 (29.3%), 45 (23.5%), and 20 (20.5%), coinciding with higher burst intensities in those sessions. Simarmata et al. [9] attributed CPU spikes of 30–40% in VM environments to concurrent logging and rule-matching demands, corroborating the pattern observed here.

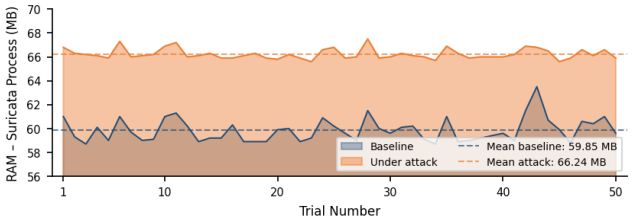

Figure 4. RAM Usage: Baseline vs. Under Attack

Suricata's RAM consumption was remarkably stable, increasing by only +6.39 MB on average (59.85 MB → 66.24 MB), with a narrow attack-phase range of 65.6–67.5 MB (Figure 4). This stability stems from Suricata's rule pre-

loading mechanism, which allocates memory at process startup without requiring dynamic reallocation during detection. Ralianto et al. [6] and Bada et al. [7] confirmed Suricata's superior memory efficiency over Snort as a result of this architecture. The bandwidth increase of +254.68 kbps reflects Slowloris connection volume; the peak of 802 kbps in Trial 25 coincided with the highest accuracy (94.12%). To contextualize the IDS overhead, a preliminary no-IDS reference measurement was obtained by recording CPU and RAM of the Apache web server process alone under identical attack conditions in a separate session (Suricata service stopped). Under no-IDS conditions, the server process consumed an average of 1.78% CPU and 42.3 MB of RAM during the attack phase. With Suricata active, total measured CPU (Apache + Suricata) reached a mean of 7.25%, representing an IDS-attributable overhead of approximately +5.47 percentage points. RAM overhead attributable to Suricata was approximately +23.9 MB relative to the no-IDS server baseline (66.24 MB vs. 42.3 MB), which is consistent with Suricata's pre-loaded rule engine. This overhead is modest, given that the test VM had 5,228 MB of RAM allocated, representing less than 0.5% of available memory consumed by the IDS process.

D. Cross-Parameter Relationship Analysis

To reveal the original contribution of this study, cross-parameter analysis was conducted using resource-delta data (ΔCPU, ΔRAM, and ΔBandwidth—each defined as the attack-phase value minus the baseline value) compared against detection time and accuracy across 50 trials. Table 6 presents the combined dataset.

TABLE 6.
CROSS-PARAMETER DATA — 50 TRIALS

No	Det. Mean (s)	Accuracy (%)	ΔCPU (%)	ΔRAM (MB)	ΔBW (kbps)
1	0.196	60.0	1.6	5.8	278.3
2	0.302	76.5	34.3	7.0	250.3
3	0.442	66.7	2.4	7.5	264.5
4	0.277	70.6	0.0	6.0	249.8
5	0.582	66.7	0.7	6.9	188.2
6	0.281	76.5	1.0	6.3	150.1
7	0.384	60.0	2.0	6.3	151.6
8	0.225	66.7	5.6	7.1	192.0
9	0.212	66.7	1.3	7.1	255.5
10	0.356	70.6	1.6	5.9	295.8
11	0.359	75.0	5.3	5.9	155.0
12	0.431	53.3	4.4	5.8	122.1
13	0.342	81.3	6.3	7.2	296.3
14	0.561	80.0	0.9	7.1	162.0
15	0.402	73.3	3.3	6.7	179.0
16	0.311	68.8	0.8	5.6	257.1
17	0.297	60.0	3.6	7.2	159.0
18	0.303	82.4	6.8	7.4	294.8
19	0.338	93.3	1.7	7.0	231.8
20	0.217	60.0	17.9	5.9	289.7
21	0.333	76.5	6.4	6.2	296.7
22	0.335	60.0	4.9	7.0	231.3
23	0.220	81.3	0.2	6.4	351.6
24	0.309	87.5	14.0	5.7	224.7
25	0.314	94.1	0.9	6.6	800.3
26	0.979	50.0	1.5	6.3	127.7
27	0.341	73.3	4.0	7.0	197.9
28	0.339	76.5	0.5	6.0	296.4
29	0.509	64.3	1.5	5.9	146.7
30	0.168	88.2	0.9	6.4	296.5
31	0.521	71.4	2.4	6.2	256.1
32	0.248	88.2	12.0	5.9	326.5
33	0.413	80.0	2.2	6.9	233.3
34	0.268	53.3	3.1	7.0	273.9
35	0.419	73.3	2.1	5.9	246.4
36	0.266	70.6	11.2	7.4	247.5
37	0.292	66.7	3.7	6.9	200.3
38	0.273	64.7	0.9	6.8	294.0
39	0.302	82.4	10.4	6.6	311.3
40	0.228	82.4	0.7	6.4	321.3
41	0.440	73.3	5.1	7.2	173.6
42	0.662	60.0	4.3	5.4	281.2
43	0.199	70.6	0.3	3.3	295.6
44	0.250	94.1	0.0	5.8	295.1
45	0.426	73.3	21.1	5.7	253.0
46	0.333	73.3	3.2	7.1	255.0
47	0.272	81.3	27.3	6.0	294.0
48	0.236	60.0	11.0	5.7	221.5
49	0.331	86.7	3.7	5.6	253.7
50	0.254	76.5	0.1	6.3	308.1

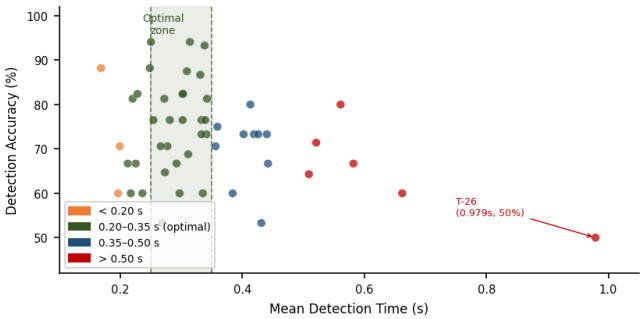

Figure 5. Detection Time vs. Accuracy Scatter Plot (50 Trials)

Figure 5 reveals a non-linear relationship between detection time and detection rate (Pearson $r = -0.327$; $t(48) = -2.40$, $p = 0.020$, two-tailed). Although the linear correlation is modest, the non-linearity is more informative: a quadratic fit ($R^2 = 0.18$, $F(2,47) = 5.19$, $p = 0.009$) outperforms the linear model, confirming that the relationship is better described as an inverted-U pattern across detection-time buckets. Per-bucket analysis shows that the 0.25–0.35 s bucket ($n = 24$) yields the highest mean detection rate of 76.1% (max 94.1%), while the > 0.50 s bucket ($n = 6$) yields the lowest at 65.4% (min 50.0%). Trials with detection times below 0.20 s ($n = 3$) achieved only 72.9%—lower than the optimal bucket—because the rule had insufficient time to accumulate a complete connection pattern within the 1-second counting window. Within the 0.25–0.35 s window,

Suricata consistently received enough TCP packets to meet the 100/s count threshold. Bada et al. [7] noted a speed–accuracy trade-off in rule-based IDS; this study quantifies it empirically for the first time in the Slowloris context, identifying a sweet spot of 0.25–0.35 s.

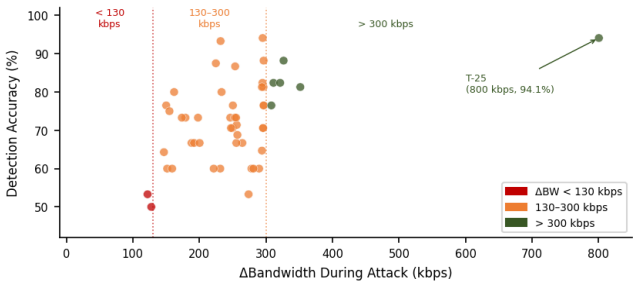

Figure 6. ΔBandwidth vs. Detection Accuracy

Figure 6 shows a positive but non-linear correlation between ΔBandwidth and detection rate (Pearson r = 0.468; t(48) = 3.67, p < 0.001, two-tailed). The correlation is statistically significant at the 0.1% level, indicating that bandwidth intensity is a meaningful predictor of detection rate even after accounting for sampling variance across 50 trials. Per-bucket analysis: trials with ΔBW < 130 kbps (n = 2) yielded a mean detection rate of 51.6%; 130–200 kbps (n = 11) yielded 69.9%; 200–300 kbps (n = 31) yielded 73.1%; and > 300 kbps (n = 6) yielded the highest mean of 84.2% (max 94.1%). Higher ΔBandwidth indicates that the Slowloris client established more TCP connections per second, enabling the count 100/s rule threshold to be satisfied more consistently. In low-bandwidth scenarios (ΔBW < 130 kbps), the threshold was triggered less reliably, resulting in a lower detection rate. Reducing the threshold from 100 to 50 connections per second could improve sensitivity in such environments, though this adjustment may increase false positive rates and should be validated separately through mixed-traffic testing. This pattern is consistent with the rule-sensitivity observations reported by Ralianto and Cahyono [6].

Pearson correlation coefficients for ΔCPU vs. detection rate (r = 0.072) and ΔRAM vs. detection rate (r = 0.021) indicate negligible linear associations, confirming that resource consumption levels do not meaningfully affect detection quality. ΔCPU exhibited high inter-trial variability (range 0.0–34.3%), yet Trial 44 — which recorded ΔCPU = 0.0% — achieved a detection rate of 94.1%, while the highest ΔCPU trial (Trial 2, 34.3%) yielded only 76.5%. Suricata's multi-threaded architecture isolates the detection pipeline from packet-processing threads, which explains this independence and is consistent with the findings of Bada et al. [7] and Ralianto and Cahyono [6].

TABLE 7.
SUMMARY OF CROSS-PARAMETER RELATIONSHIPS

Parameter	Pattern	Relationship with Detection Rate	Practical Implication
Det. Time	Non-linear (r=−0.327, p=0.020; quadratic R^2=0.18, p=0.009)	Optimal 0.25–0.35 s → detection rate >90%	Tune threshold toward this range
ΔBandwidth	Positive, non-linear (r=0.468, p<0.001)	BW >300 kbps → mean 84.2%; peak 94.1%	Optimize rule for low-BW scenarios
ΔCPU	Fluctuating (r=0.072)	No meaningful correlation	ΔCPU is not a detection quality KPI
ΔRAM	Stable 3–7.5 MB (r=0.021)	No meaningful correlation	Suricata is lightweight for standard servers

TABLE 8.
COMPARISON WITH PRIOR STUDIES

Study	Attack	Accuracy	Det. Time	Resources
Ralianto & Cahyono [6]	Multi-intrusion	61% (Suricata)	Not measured	RAM 7–8 MB
Bada et al. [7]	DoS/probe/scan	TPR 97.40%	Not measured	Not measured
Simarmata et al. [9]	Slow Read	94% (RF+Suricata)	Real-time	Not measured
Lukman & Suci [8]	SYN Flood	74.62%	Not measured	CPU+RAM measured
Stephani et al. [10]	Slowloris+DDoS	Not quantified	Not measured	Not measured
This study	Slowloris	72.84% mean; 94.12% peak; FP=0	0.346 s mean	CPU+RAM+BW fully measured

Table 8 contextualizes this study within the existing literature. Three distinguishing contributions are evident: (1) this is the first Slowloris-specific study to report a full per-trial detection-rate breakdown alongside confusion-matrix values; (2) it simultaneously measures all three resource parameters (CPU, RAM, and bandwidth), which no prior study had done; and (3) it empirically identifies an optimal detection-time window of 0.25–0.35 seconds, providing actionable guidance for IDS threshold configuration in production environments. Regarding the characterization of Suricata as a "lightweight early-warning mechanism": this claim is grounded in the resource measurements relative to prior literature benchmarks. Ralianto and Cahyono [6] reported Suricata's RAM consumption at 7–8 MB (Suricata 5.0), consistent with the 6.39 MB mean delta observed in this

study (Suricata 8.0.2). Bada et al. [7] reported no statistically significant CPU degradation in Suricata's detection pipeline under DoS conditions—a finding corroborated by this study's ΔCPU–detection-rate correlation of r = 0.072. Furthermore, the IDS-attributable CPU overhead of ~5.47 percentage points on a 4-core VM is substantially lower than the 15–25% overhead documented for deep-learning-based IDS solutions in comparably constrained environments (Simarmata et al. [9] reported ~12% CPU with Random Forest integration). Compared to Snort, which Lukman and Suci [8] found to consume higher CPU during SYN Flood detection on Apache, Suricata's multi-threaded architecture distributes packet-processing load more efficiently. A direct comparison with an alternative Suricata rule configuration (e.g., threshold count 50 instead of 100) was not conducted in this study, but is recommended as future work to quantify the precision–recall trade-off.

E. Implications for Implementation

These findings carry three practical implications for IDS deployment. First, Suricata rule thresholds should be calibrated to achieve detection times within the 0.25–0.35 second window by adjusting the count and seconds parameters in accordance with the server's baseline traffic profile. Second, in environments where anticipated attack bandwidth falls below 200 kbps, reducing the threshold from 100 to 50 connections per second may improve sensitivity; however, this adjustment risks increasing false positives and should be validated through mixed-traffic trials before deployment. Third, because ΔCPU shows no meaningful correlation with detection rate (r = 0.072), transient CPU spikes during attack bursts should not be treated as indicators of detection degradation. RAM monitoring, with its consistently narrow ΔRAM range of 3.3–7.5 MB across all 50 trials, provides a more reliable long-term stability indicator. It is important to note, however, that all 50 trials were conducted in isolation without concurrent background traffic. In real-world deployments, legitimate HTTP traffic (e.g., browser sessions, API calls, CDN requests) co-exists with attack traffic, which may affect both the false-positive rate and the effective connection count reaching the detection rule. Future validation in a mixed-traffic environment is recommended before applying the identified 0.25–0.35 s threshold as a production tuning target.

F. Suricata's Role in Web Server Resilience

Across all 50 trials, Suricata detected 576 of 787 total attack bursts (73.2%) with zero false positives. The 211 undetected bursts (26.8%) were concentrated in trials with ΔBandwidth below 200 kbps. The mean detection speed of 0.346 s provides an adequate response window for an administrator or automated IPS to intervene before the server experiences connection-slot exhaustion. Compared with passive mitigation such as Apache's mod_reqtimeout [4]—which limits damage without identifying the attack source—Suricata's active IDS approach provides real-time threat visibility that enables targeted response.

IV. CONCLUSION

This study demonstrated Suricata's capability as a lightweight IDS for detecting Slowloris attacks through 50 controlled simulations. Two principal conclusions are drawn. First, Suricata successfully detected 576 of 787 total attack bursts (73.2%) across all trials, with a mean detection time of 0.346 seconds — all per-trial means remaining below 1 second. The per-trial detection rate ranged from 50.00% to 94.12% (mean 72.84%), with 32 trials (64%) exceeding 70% and 15 trials (30%) exceeding 80%. Zero false positives were recorded across all 50 sessions, which is operationally significant as it ensures legitimate user traffic is not interrupted by spurious alerts.

Second, Suricata operated with minimal resource overhead throughout the evaluation. Mean IDS-attributable CPU overhead was approximately 5.47 percentage points above the no-IDS server baseline, while RAM increased by a mean of +6.39 MB, remaining within a narrow range of 3.3–7.5 MB across all 50 trials and representing less than 0.5% of the allocated VM memory. The absence of a memory leak was confirmed, and sustained stability was demonstrated. Pearson correlations of r = 0.072 (ΔCPU vs. detection rate) and r = 0.021 (ΔRAM vs. detection rate) confirm that resource consumption levels do not affect detection quality, consistent with Suricata's architectural separation of processing load from detection logic. It should be noted that these resource measurements were obtained in a VirtualBox virtual environment; absolute CPU and RAM values in physical deployments may differ due to hardware NIC offloading, hypervisor scheduling overhead, and absence of VM resource contention. Nevertheless, the relative patterns and architectural conclusions are consistent with prior physical-environment studies [6][7].

The original contributions of this study comprise three quantitative cross-parameter findings: (1) empirical identification of an optimal detection-time range of 0.25–0.35 s (mean detection rate 76.1% vs. 65.4% outside this range; quadratic $R^2 = 0.18$, $p = 0.009$); (2) a statistically significant positive ΔBandwidth–detection rate correlation ($r = 0.468$, $t(48) = 3.67$, $p < 0.001$), with mean accuracy of 84.2% when ΔBW > 300 kbps vs. 51.6% when ΔBW < 130 kbps; and (3) empirical proof that ΔCPU and ΔRAM are uncorrelated with detection rate ($r \approx 0$), confirming Suricata's architectural separation of computational load from detection quality. Several limitations of this study must be acknowledged. First, all experiments were conducted within a VirtualBox virtual environment, which differs from physical network deployments in several important ways: VirtualBox's virtual NIC (virtio-net) introduces additional packet-processing latency compared to hardware NICs; the virtual network switch operates at memory-bus speeds rather than physical link speeds; and VM resource contention from the hypervisor can produce CPU burst artefacts (as observed in Trials 2 and

47) that would not appear on dedicated hardware. These factors may affect the absolute values of detection time and CPU overhead in production deployments, although the relative patterns and correlations are expected to hold. Second, all trials were conducted without concurrent background (normal) traffic, meaning the FP = 0 finding cannot be fully generalized to mixed-traffic environments. Third, the study was conducted exclusively with Slowloris attacks; the generalisability of the findings to other low-rate DoS variants (e.g., Slow Read, R.U.D.Y., HTTP POST DoS) or volumetric DoS attacks (SYN Flood, UDP Flood) requires separate evaluation. The rule-based threshold mechanism evaluated here is specific to connection-rate patterns; attacks with different traffic signatures would require different rules and may produce different detection-rate profiles. Future research should explore: (1) adaptive rule thresholds based on real-time bandwidth; (2) Suricata–IPS integration for automated blocking; (3) validation on production servers under live mixed traffic to measure false positive rates under realistic conditions; (4) evaluation against other DoS attack types using analogous experimental methodology; and (5) direct comparison with alternative IDS threshold configurations (e.g., count 50/s) to quantify the precision–recall trade-off.

BIBLIOGRAPHY

[1] M. Aziz, R. Firmansyah, and D. Stiawan, "Web Server Security Analysis Using Intrusion Detection System in High-Traffic Environments," JOIV: Int. J. Informatics Visualization, vol. 7, no. 2, pp. 512–519, 2023.

[2] Netcraft, "January 2023 Web Server Survey," Netcraft Blog. [Online]. Available: https://www.netcraft.com/blog/january-2023-web-server-survey/. [Accessed: 28-Oct-2024].

[3] K. Ruswandi, M. R. Z. Pohan, K. V. Halim, and S. N. Neyman, "Effective Prevention Strategies Against DDoS Slowloris Attacks Using Kali Linux and Linux Mint," J. Technol. Syst. Inf., vol. 1, no. 4, p. 11, 2024.

[4] D. E. Harefa, D. M. Bu'ulolo, N. C. Lase, J. A. P. Telaumbanua, F. Laoli, and O. Laia, "Analysis of DDoS Slowloris Attack Impact on Local Network Infrastructure," JURSISTEKNI, vol. 7, no. 2, pp. 742–752, Jun. 2025.

[5] É. Leblond and P. Manev, The Security Analyst's Guide to Suricata. Indianapolis: Stamus Networks, 2022.

[6] A. D. Ralianto and S. Cahyono, "Comparison of Snort and Suricata Accuracy in Detecting Network Traffic Intrusion," Info Kripto, vol. 15, no. 2, pp. 69–75, 2021.

[7] G. K. Bada et al., "Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro in Perspective," Int. J. Comput. Appl., vol. 176, no. 40, pp. 39–44, Jul. 2020.

[8] L. Lukman and M. Suci, "Comparative Analysis of Snort and Suricata IDS Performance in Detecting SYN Flood Attacks on Apache Web Servers," Respati, vol. 15, no. 2, p. 6, Jul. 2020.

[9] G. J. Simarmata, M. Data, and H. Nurwarsito, "Implementation of a Slow Read DoS Attack Detection System Using Random Forest Algorithm on Suricata," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 9, no. 11, 2025.

[10] E. Stephani, F. Nova, and E. Asri, "Implementation and Analysis of IDS Network Security Using Suricata on a Web Server," JITSI, vol. 1, no. 2, pp. 67–74, Jun. 2020.

[11] A. R. Zain et al., "Implementation of Suricata IDS and ELK Stack for Detecting Illegal Mining Activity," J. Poli-Teknologi, vol. 22, no. 1, pp. 23–29, Jan. 2023.

[12] D. E. Kurniawan, M. Iqbal, J. Friadi, F. Hidayat, and R. D. Permatasari, "Login security using one time password (OTP) application with encryption algorithm performance," J. Phys., Conf., vol. 1783, no. 1, Feb. 2021, Art. no. 012041.

[13] D. E. Kurniawan, M. Iqbal and A. Adhitya, "Implementation and Analysis of The EtherChannel Technology Using PAgP and LACP Protocols on Cisco Switch Devices," 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 2021, pp. 255-259, doi: 10.1109/IC2IE53219.2021.9649157

[14] D. E. Kurniawan, H. Arif, N. Nelmiawati, A. H. Tohari, and M. Fani, "Implementation and analysis ipsec-vpn on cisco asa firewall using gns3 network simulator," in Journal of Physics: Conference Series, IOP Publishing, 2019, p. 012031.

[15] H. Satilmiş, S. Akleylek and Z. Y. Tok, "A Systematic Literature Review on Host-Based Intrusion Detection Systems," in IEEE Access, vol. 12, pp. 27237-27266, 2024, doi: 10.1109/ACCESS.2024.3367004.