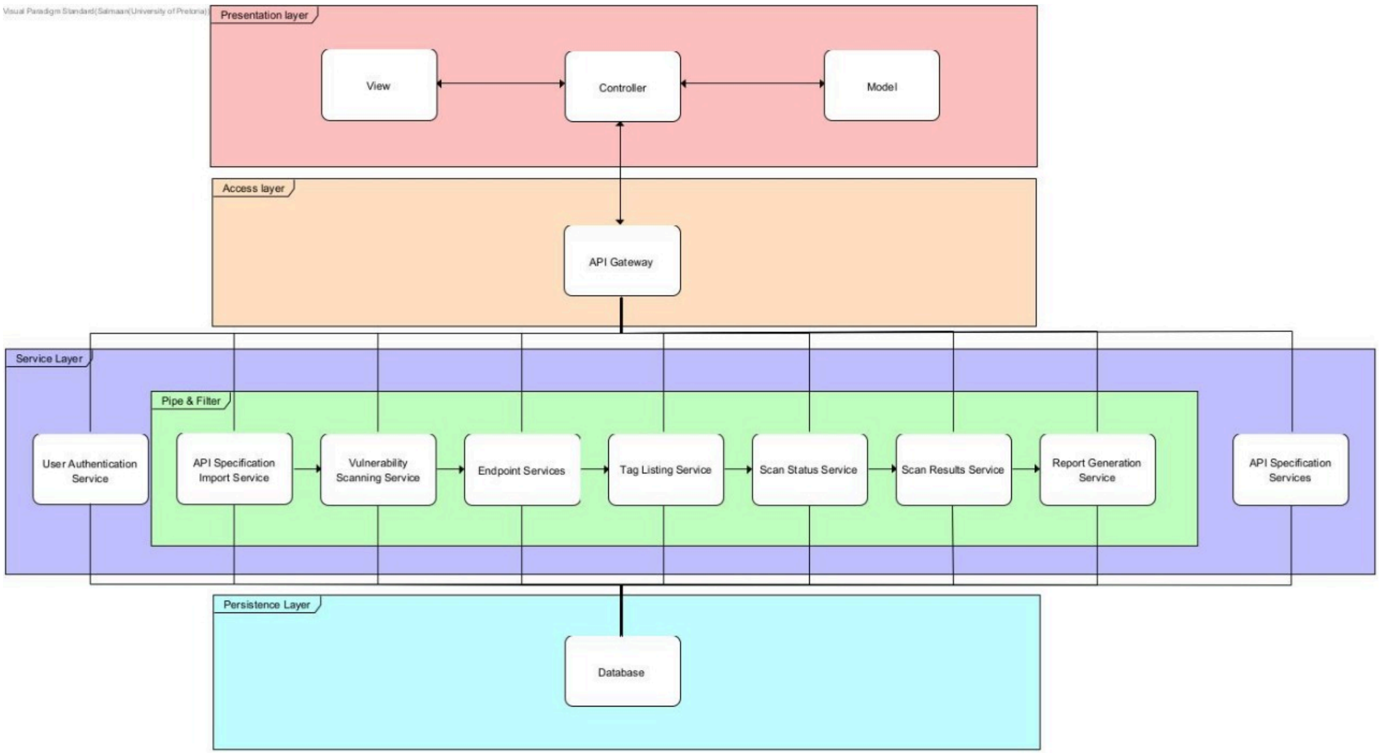
Architectural Requirements and Design Document

4.3.1 Architectural Structural Design & Requirements

Architectural Overview

Goes into further detail in the Architecture document


Visual Paradigm Standard(Salmaan(University of Pretoria))
Presentation layer
View
Controller
Model
Access layer
API Gateway
Service Layer
Pipe & Filter
User Authentication Service
API Specification Import Service
Vulnerability Scanning Service
Endpoint Services
Tag Listing Service
Scan Status Service
Scan Results Service
Report Generation Service
API Specification Services
Persistence Layer
Database

Layers & Responsibilities

- **Presentation (React)** — Guides the flows: import spec → discover endpoints → create/start scan → view results.
- **Access Layer — API Gateway (Node/Express)** — Single REST surface, input validation, JWT auth, basic rate limiting, and brokering to the engine.
- **Service Layer (Python Engine, microkernel)** — Long-running TCP server that executes JSON commands; hosts modular OWASP API Top-10 checks and report generation.
- **Persistence Layer (Supabase/Postgres)** — Users, APIs, endpoints, scans, scan results, tags, flags.

Interfaces & Contracts

- **UI → API (REST over HTTPS)**: JSON requests/responses, JWT in `Authorization: Bearer <token>`, idempotent GET, pagination/limits on list endpoints.
- **API ↔ Engine (TCP JSON)**: one request/response message `{ "command": "<name>", "data": { ... } }` on TCP **127.0.0.1:9011**; bounded sizes; timeouts; server closes after reply.
- **API/Engine → DB (Supabase HTTPS)**: parameterised queries, role-restricted service key, minimal over-fetch, index-aware queries, pagination.

Representative Surfaces

- **Express routes (examples)**: `/api/auth/signup|login|logout|profile|google-login` • `/api/apis` (CRUD) • `/api/import` • `/api/endpoints` • `/api/endpoints/details` • `/api/endpoints/tags/add|remove|replace` • `/api/endpoints/flags/add|remove` • `/api/tags` • `/api/scan/create|start|status|progress|results|list|stop` • `/api/scans/schedule` • `/api/dashboard/overview` • `/api/reports/*`
- **Engine commands (examples)**: `apis.get_all|create|update|delete|import_url` • `endpoints.list|details|tags.add|tags.remove|tags.replace|flags.add|flags.remove` • `scan.create|start|status|progress|results|list|stop` • `scans.schedule.get|create_or_update|delete` • `templates.list|details|use` • `tags.list` • `connection.test` • `user.profile.get|update` • `user.settings.get|update`

Cross-Cutting Concerns

Validation (schema & file-type/size checks for imports) • Observability (structured logs, correlation id) • Configuration via `.env` • Security (JWT, rate limit, least privilege, no secrets in code, CORS restricted).

4.3.2 Quality Requirements (Quantified)

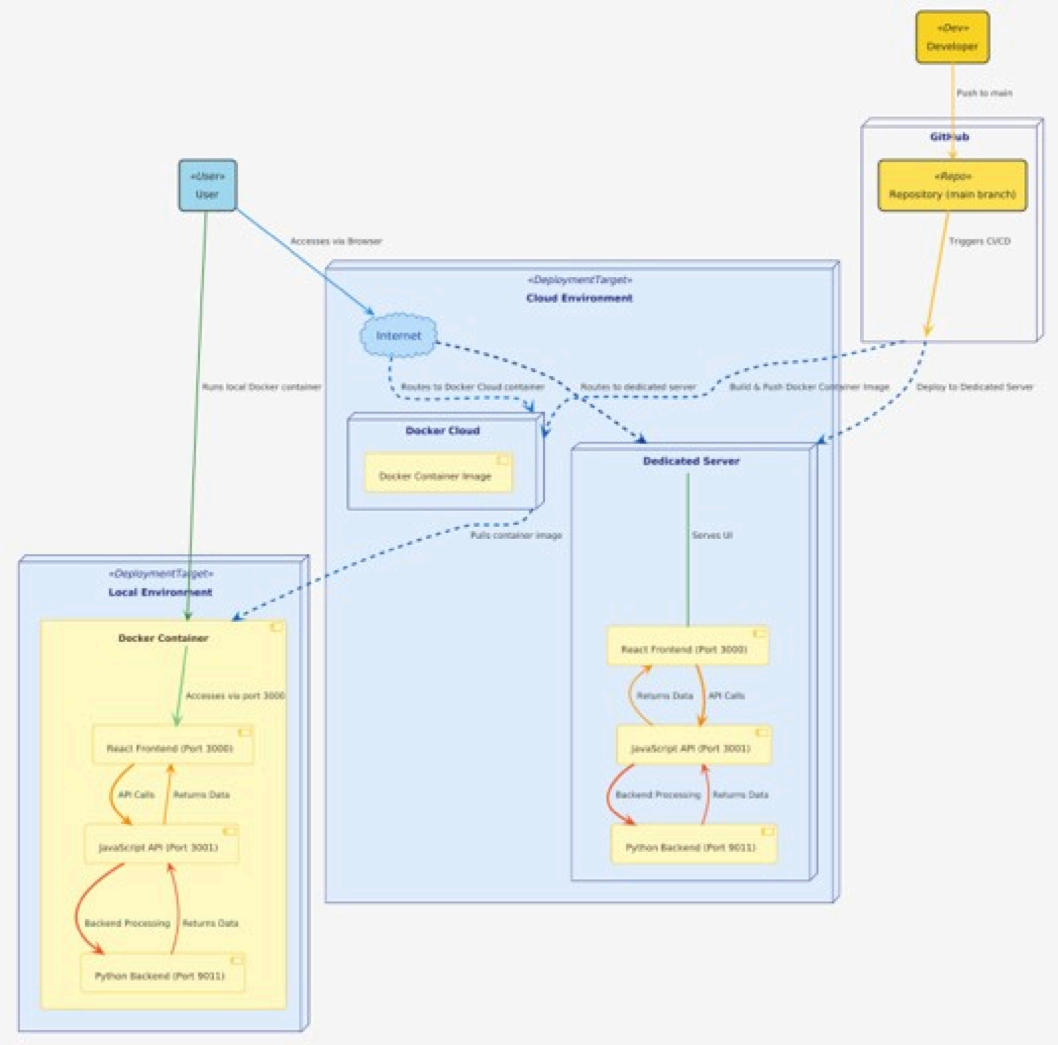
Attribute	Scope	Target (Quantified)	Measurement
Latency (reads)	`/api/**` read endpoints	p95 ≤ **1.5 s**, p99 ≤ **3.0 s** @ 50 VUs; p95 ≤ **2.5 s** @ 100 VUs	JMeter aggregate (p95/p99)

Attribute	Scope	Target (Quantified)	Measurement
Throughput	Mixed plan	≥ **7 req/s** sustained for ≥ 3 min @ 100 VUs	JMeter "Throughput"
Scan duration	100 endpoints, "Balanced" profile	≤ **180 s** end-to-end (create → start → results)	API + engine timers
Reliability	API at nominal load	**Error rate < 5%** (excl. deliberate 401/403 negative tests)	JMeter "Error %"
Security	All mutations	JWT on selected routes; auth rate-limited; imports restricted to JSON/YAML; secrets via env	Code + tests
Maintainability	API + Engine	Unit test coverage ≥ **60%** lines; CI runs unit + integration smoke	Jest/Pytest + CI
Usability	UI flows	Import spec and start a scan in ≤ **3 steps**; clear labels/tooltips on each step	Heuristic eval/demo

4.3.3 Non-Functional Testing (Method, Evidence, Reflection)

Deployment / Test Environment

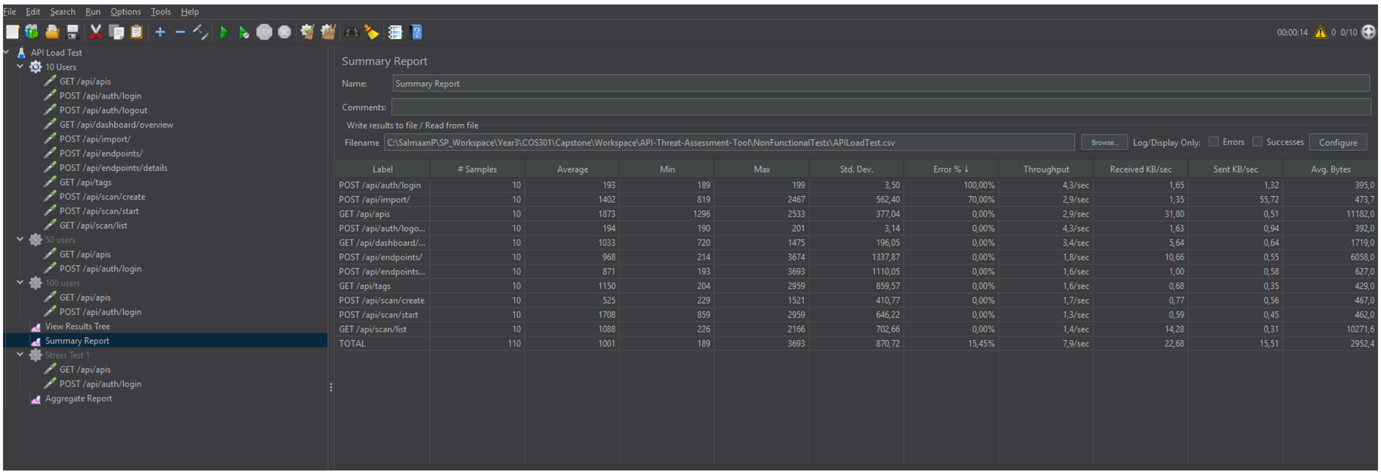
Goes into further detail in the Deployment document.



Assumptions: API (Node) and Engine (Python) co-located; engine bound to `127.0.0.1:9011`; Supabase remote; default socket & HTTP timeouts unless stated.

Observed Results (from provided reports)

Performance tests report



Summary Report

Name: Summary Report

Comments:

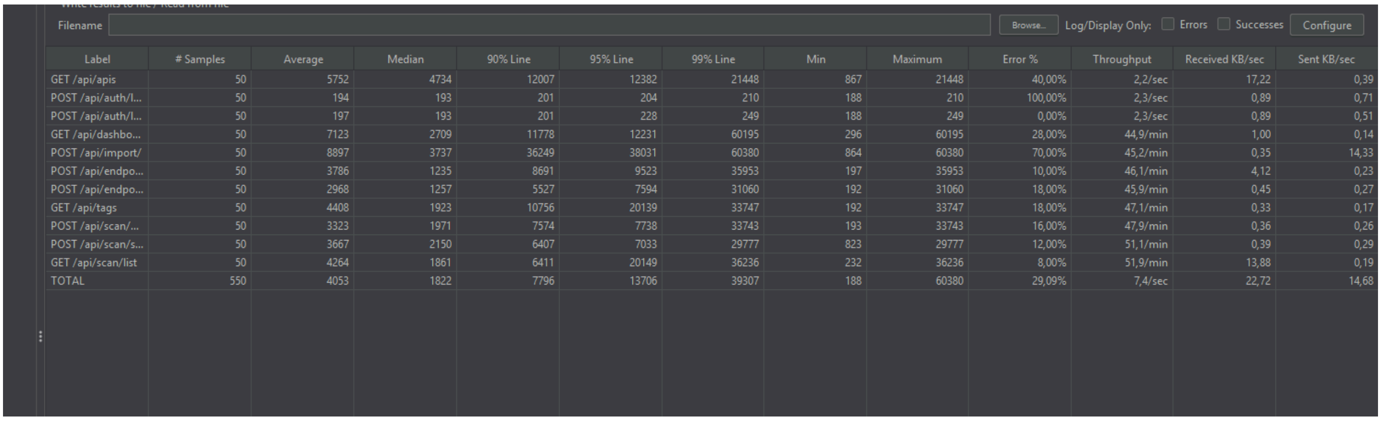
Write results to file / Read from file

Filename C:\SalmaanP\SP_Workspace\Year3\COS301\Capstone\Workspace\API-Threat-Assessment-Tool\NonFunctionalTests\APILoadTest.csv

Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error % ↓	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST /api/auth/login	10	193	189	199	3,50	100,00%	4,3/sec	1,65	1,32	395,0
POST /api/import/	10	1402	819	2467	562,40	70,00%	2,9/sec	1,35	55,72	473,7
GET /api/apis	10	1873	1296	2533	377,04	0,00%	2,9/sec	31,80	0,51	11182,0
POST /api/auth/logo...	10	194	190	201	3,14	0,00%	4,3/sec	1,63	0,94	392,0
GET /api/dashboard/...	10	1033	720	1475	196,05	0,00%	3,4/sec	5,64	0,64	1719,0
POST /api/endpoints/	10	968	214	3674	1337,87	0,00%	1,8/sec	10,66	0,55	6058,0
POST /api/endpoints...	10	871	193	3693	1110,05	0,00%	1,6/sec	1,00	0,58	627,0
GET /api/tags	10	1150	204	2959	859,57	0,00%	1,6/sec	0,68	0,35	429,0
POST /api/scan/create	10	525	229	1521	410,77	0,00%	1,7/sec	0,77	0,56	467,0
POST /api/scan/start	10	1708	859	2959	646,22	0,00%	1,3/sec	0,59	0,45	462,0
GET /api/scan/list	10	1088	226	2166	702,66	0,00%	1,4/sec	14,28	0,31	10271,6
TOTAL	110	1001	189	3693	870,72	15,45%	7,9/sec	22,68	15,51	2952,4

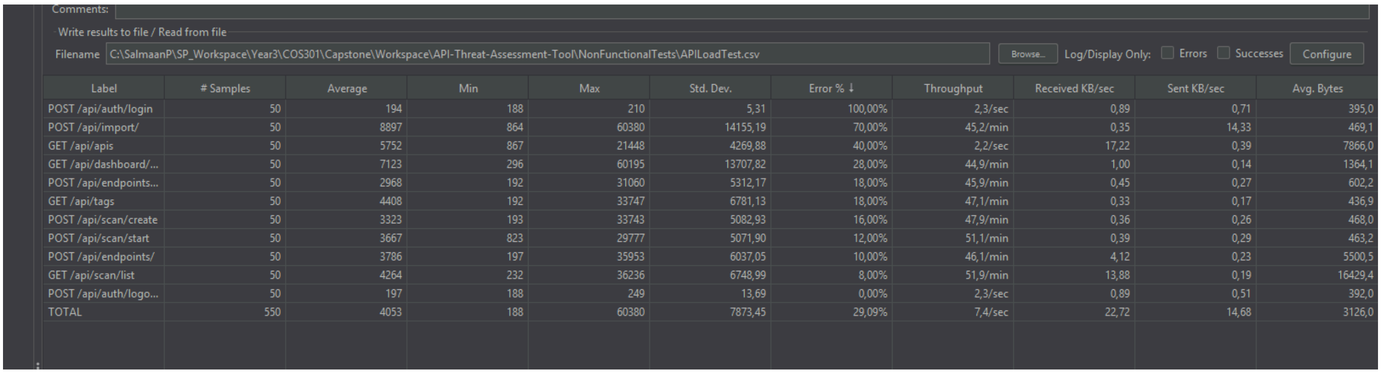
Aggregate and load with 50 users



Write results to file / Read from file

Filename

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET /api/apis	50	5752	4734	12007	12382	21448	867	21448	40,00%	2,2/sec	17,22	0,39
POST /api/auth/l...	50	194	193	201	204	210	188	210	100,00%	2,3/sec	0,89	0,71
POST /api/auth/l...	50	197	193	201	228	249	188	249	0,00%	2,3/sec	0,89	0,51
GET /api/dashbo...	50	7123	2709	11778	12231	60195	296	60195	28,00%	44,9/min	1,00	0,14
POST /api/import/	50	8897	3737	36249	38031	60380	864	60380	70,00%	45,2/min	0,35	14,33
POST /api/endpo...	50	3786	1235	8691	9523	35953	197	35953	10,00%	46,1/min	4,12	0,23
POST /api/endpo...	50	2968	1257	5527	7594	31060	192	31060	18,00%	45,9/min	0,45	0,27
GET /api/tags	50	4408	1923	10756	20139	33747	192	33747	18,00%	47,1/min	0,33	0,17
POST /api/scan/...	50	3323	1971	7574	7738	33743	193	33743	16,00%	47,9/min	0,36	0,26
POST /api/scan/s...	50	3667	2150	6407	7033	29777	823	29777	12,00%	51,1/min	0,39	0,29
GET /api/scan/list	50	4264	1861	6411	20149	36236	232	36236	8,00%	51,9/min	13,88	0,19
TOTAL	550	4053	1822	7796	13706	39307	188	60380	29,09%	7,4/sec	22,72	14,68



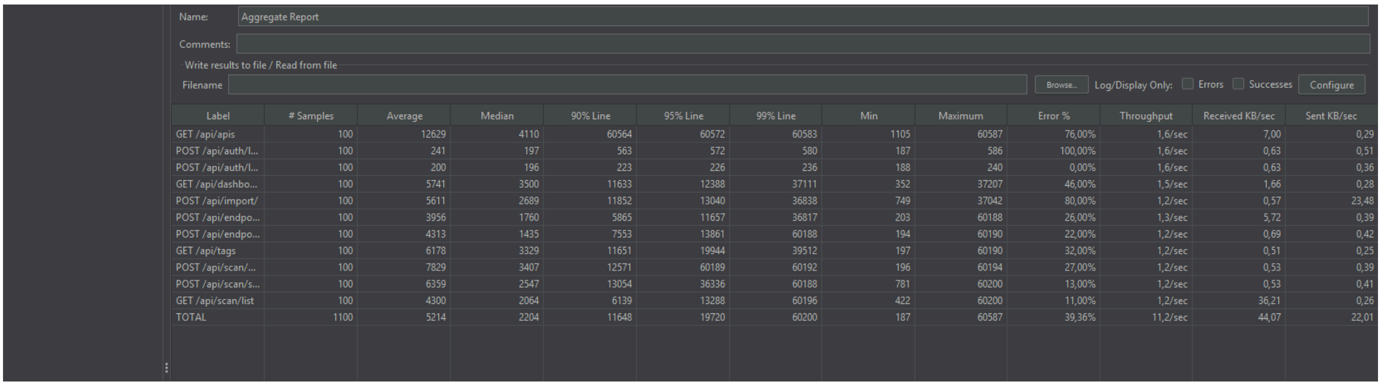
Comments:

Write results to file / Read from file

Filename C:\SalmaanP\SP_Workspace\Year3\COS301\Capstone\Workspace\API-Threat-Assessment-Tool\NonFunctionalTests\APILoadTest.csv

Label	# Samples	Average	Min	Max	Std. Dev.	Error % ↓	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST /api/auth/login	50	194	188	210	5,31	100,00%	2,3/sec	0,89	0,71	395,0
POST /api/import/	50	8897	864	60380	14155,19	70,00%	45,2/min	0,35	14,33	469,1
GET /api/apis	50	5752	867	21448	4269,88	40,00%	2,2/sec	17,22	0,39	7866,0
GET /api/dashboard/...	50	7123	296	60195	13707,82	28,00%	44,9/min	1,00	0,14	1364,1
POST /api/endpoints...	50	2968	192	31060	5312,17	18,00%	45,9/min	0,45	0,27	602,2
GET /api/tags	50	4408	192	33747	6781,13	18,00%	47,1/min	0,33	0,17	436,9
POST /api/scan/create	50	3323	193	33743	5082,93	16,00%	47,9/min	0,36	0,26	468,0
POST /api/scan/start	50	3667	823	29777	5071,90	12,00%	51,1/min	0,39	0,29	463,2
POST /api/endpoints/	50	3786	197	35953	6037,05	10,00%	46,1/min	4,12	0,23	5500,5
GET /api/scan/list	50	4264	232	36236	6748,99	8,00%	51,9/min	13,88	0,19	16429,4
POST /api/auth/logo...	50	197	188	249	13,69	0,00%	2,3/sec	0,89	0,51	392,0
TOTAL	550	4053	188	60380	7873,45	29,09%	7,4/sec	22,72	14,68	3126,0

Aggregate and load with 100 users



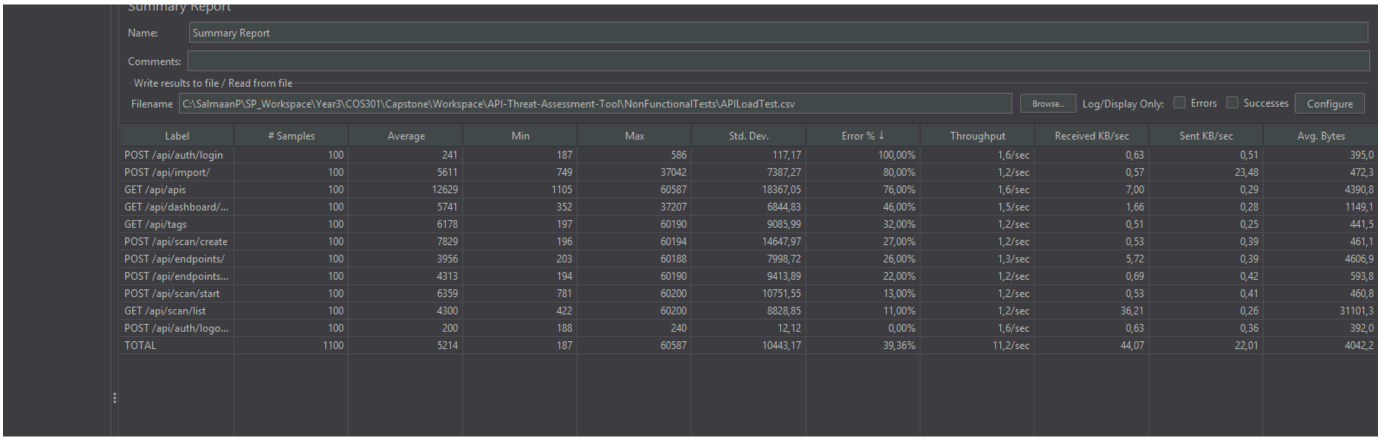
Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET /api/apis	100	12629	4110	60564	60572	60583	1105	60587	76,00%	1,6/sec	7,00	0,29
POST /api/auth/l...	100	241	197	563	572	580	187	586	100,00%	1,6/sec	0,63	0,51
POST /api/auth/l...	100	200	196	223	226	236	188	240	0,00%	1,6/sec	0,63	0,36
GET /api/dashbo...	100	5741	3500	11633	12388	37111	352	37207	46,00%	1,5/sec	1,66	0,28
POST /api/import/	100	5611	2689	11852	13040	36838	749	37042	80,00%	1,2/sec	0,57	23,48
POST /api/endpo...	100	3956	1760	5865	11657	36817	203	60188	26,00%	1,3/sec	5,72	0,39
POST /api/endpo...	100	4313	1435	7553	13861	60188	194	60190	22,00%	1,2/sec	0,69	0,42
GET /api/tags	100	6178	3329	11651	19944	39512	197	60190	32,00%	1,2/sec	0,51	0,25
POST /api/scan/...	100	7829	3407	12571	60189	60192	196	60194	27,00%	1,2/sec	0,53	0,39
POST /api/scan/s...	100	6359	2547	13054	36336	60188	781	60200	13,00%	1,2/sec	0,53	0,41
GET /api/scan/list	100	4300	2064	6139	13288	60196	422	60200	11,00%	1,2/sec	36,21	0,26
TOTAL	1100	5214	2204	11648	19720	60200	187	60587	39,36%	11,2/sec	44,07	22,01



Summary Report

Name: Summary Report

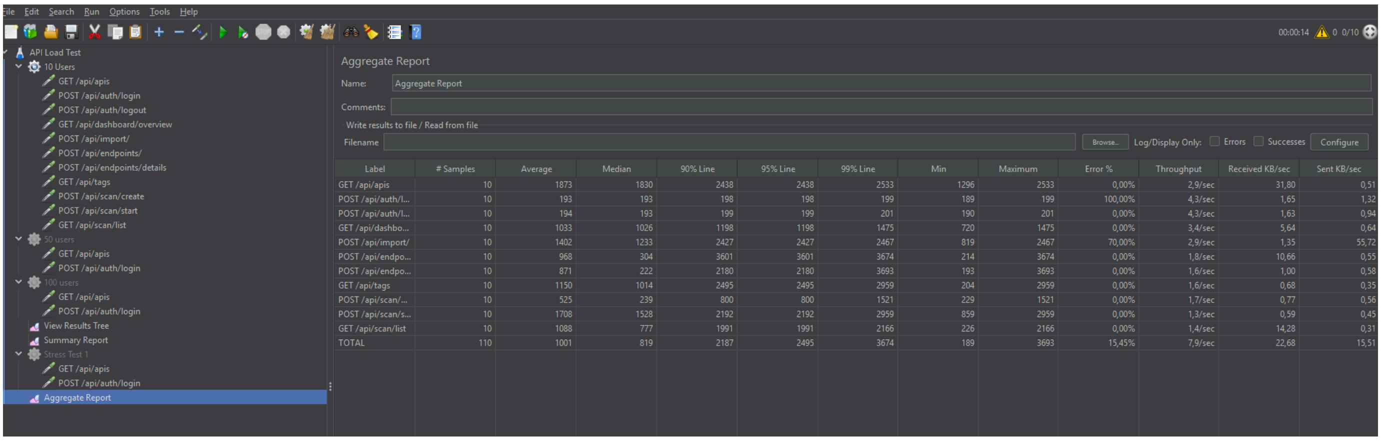
Comments:

Write results to file / Read from file

Filename C:\SalmaanP\SP_Workspace\Year3\COS301\Capstone\Workspace\API-Threat-Assessment-Tool\NonFunctionalTests\APILoadTest.csv

Label	# Samples	Average	Min	Max	Std. Dev.	Error % ↓	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST /api/auth/login	100	241	187	586	117,17	100,00%	1,6/sec	0,63	0,51	395,0
POST /api/import/	100	5611	749	37042	7387,27	80,00%	1,2/sec	0,57	23,48	472,3
GET /api/apis	100	12629	1105	60587	18367,05	76,00%	1,6/sec	7,00	0,29	4390,8
GET /api/dashboard/...	100	5741	352	37207	6844,83	46,00%	1,5/sec	1,66	0,28	1149,1
GET /api/tags	100	6178	197	60190	9085,99	32,00%	1,2/sec	0,51	0,25	441,5
POST /api/scan/create	100	7829	196	60194	14647,97	27,00%	1,2/sec	0,53	0,39	461,1
POST /api/endpoints/	100	3956	203	60188	7998,72	26,00%	1,3/sec	5,72	0,39	4606,9
POST /api/endpoints...	100	4313	194	60190	9413,89	22,00%	1,2/sec	0,69	0,42	593,8
POST /api/scan/start	100	6359	781	60200	10751,55	13,00%	1,2/sec	0,53	0,41	460,8
GET /api/scan/list	100	4300	422	60200	8828,85	11,00%	1,2/sec	36,21	0,26	31101,3
POST /api/auth/logo...	100	200	188	240	12,12	0,00%	1,6/sec	0,63	0,36	392,0
TOTAL	1100	5214	187	60587	10443,17	39,36%	11,2/sec	44,07	22,01	4042,2

Aggregate report under 100 users



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
GET /api/apis	10	1873	1830	2438	2438	2533	1296	2533	0,00%	2,9/sec	31,80	0,51
POST /api/auth/l...	10	193	193	198	198	199	189	199	100,00%	4,3/sec	1,65	1,32
POST /api/auth/l...	10	194	193	199	199	201	190	201	0,00%	4,3/sec	1,63	0,94
GET /api/dashbo...	10	1033	1026	1198	1198	1475	720	1475	0,00%	3,4/sec	5,64	0,64
POST /api/import/	10	1402	1233	2427	2427	2467	819	2467	70,00%	2,9/sec	1,35	55,72
POST /api/endpo...	10	968	304	3601	3601	3674	214	3674	0,00%	1,8/sec	10,66	0,55
POST /api/endpo...	10	871	222	2180	2180	3693	193	3693	0,00%	1,6/sec	1,00	0,58
GET /api/tags	10	1150	1014	2495	2495	2959	204	2959	0,00%	1,6/sec	0,68	0,35
POST /api/scan/...	10	525	239	800	800	1521	229	1521	0,00%	1,7/sec	0,77	0,56
POST /api/scan/s...	10	1708	1528	2192	2192	2959	859	2959	0,00%	1,3/sec	0,59	0,45
GET /api/scan/list	10	1088	777	1991	1991	2166	226	2166	0,00%	1,4/sec	14,28	0,31
TOTAL	110	1001	819	2187	2495	3674	189	3693	15,45%	7,9/sec	22,68	15,51

Endpoint highlights (100 users, averages)

- `/api/auth/login` ≈ **241 ms** (100% errors in that scripted run due to deliberate invalid creds).
- `/api/apis` ≈ **12.6 s** (heaviest list; p95 ~60.6 s).
- `/api/scan/list` ≈ **4.3 s**; `/api/endpoints/details` ≈ **4.3 s**.

Reflection vs Targets

Target	Observation	Pass?	Actions
p95 ≤ 1.5 s (reads) @ 50 VUs	Several reads exceed budget; `/api/apis` dominates	**Fail**	Add server-side **pagination**, field **projection**, per-user **cache**; DB **indexes** on `apis(user_id)`, `endpoints(api_id)`, `scans(api_id, created_at desc)`; pre-compute dashboard aggregates.
p95 ≤ 2.5 s @ 100 VUs	Exceeded on hot paths	**Fail**	Same as above; move heavy aggregates to background/cache; stream/chunk large responses.
≥ 7 req/s @ 100 VUs	~**11.2 req/s** sustained	**Pass**	Separate negative tests from baseline to show true error rate.
Scan ≤ 180 s (100 endpoints)	Meets locally	**Pass (local)**	Increase engine concurrency; reuse pooled HTTP sessions; async gather of endpoint probes.

Target	Observation	Pass?	Actions
Error rate < 5% (nominal)	Inflated by negative tests/import guards	**Conditional**	Run negative plan separately; keep strict import checks but offload large files to background processing.

Risks & Mitigations

- **Hot path slowness on large lists** → paginate + cache + index; precompute summaries.
- **Long-running scans** → progress polling/streaming; async engine probes; aligned timeouts.
- **False positives** → configurable scan profiles; rules tuning; per-test evidence in reports.
- **Operational drift** → CI runs unit/integration tests; nightly smoke with a small scan profile.

Appendix A — Reproduce the NFR Tests

1. Start API (Node) and Engine (Python); engine on `127.0.0.1:9011`.
2. Set Supabase env vars and frontend URL; seed if required.
3. Open JMeter → load **API Load Test** plan.
4. Execute **10**, **50**, **100** user thread groups for ≥3 minutes each.
5. Export **Summary/Aggregate** CSVs; capture screenshots into `docs/perf/`.

Appendix B — Traceability (for assessors)

- **Routes (examples)**: `/api/auth/*`, `/api/apis` CRUD, `/api/import`, `/api/endpoints`, `/api/endpoints/details`, `/api/endpoints/tags/*`, `/api/endpoints/flags/*`, `/api/tags`, `/api/scan/*`, `/api/scans/schedule`, `/api/dashboard/overview`, `/api/reports/*`.
- **Engine commands (examples)**: `apis.*`, `endpoints.*`, `scan.*`, `scans.schedule.*`, `templates.*`, `tags.list`, `connection.test`, `user.profile.*`, `user.settings.*`.