iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80531

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AssessArc: System Architecture and Methodology of a Mobile Workbench for Authorized Security Assessment

Prof. Rushikesh S. Bhalerao, Vishal Satle, Satyam Katkade, Tanvi Sangale, Tanishq Medhane
Department of Information Technology SVIT, Nashik, Maharashtra, India

Abstract: ***WepresentAssessArc,anAndroidapplicationstruc-tured as a modular workbench for authorized security as-sessment, oriented toward analysts who lack a conventional workstation:thephonebecomestheprimaryreconnaissanceand manual-testing surface for discovering and documenting vul-nerabilities in scope. The system integrates rules-of-engagement (RoE) recording, vulnerability-assessment (VA) reconnaissance, manual penetration-testing (PT) aids, offline utilities, and on-device review of saved scan artifacts. Its architecture is delib-erately operator-driven: functional areas share governance con-cepts (gated network activity for the highest-risk emitters, local textual evidence with optional cryptographic fingerprints, and manualhand off betweentools)insteadofamonolithicautomated pipeline that could obscure intent or exceed scope silently. We articulate (i) the RoE lifecycle and its selective enforcement model,(ii)methodologicalpatternsofthefourVAmodules and twelve PT capabilities, (iii) offline utilities and the report repository including deduplication semantics, (iv) cross-cutting workflowsandpersistencetrade-offs,and(v)limitationsrelevant to research reproducibility, security properties of on-device stor-age, and responsible use. The contribution is architectural and methodological: a coherent field-oriented decomposition aligned with mainstream PT/VA frameworks, suitable for adaptation, extension, or formal verification in future work.***

Index Terms: ***Mobile security, penetration testing, vulnera-bility assessment, rules of engagement, Android, security work-bench, reconnaissance methodology, local evidence.***

I. INTRODUCTION

MOBILEdevicesareincreasinglyviableasfieldkits whenlaptopsareimpractical,andformanystudents, practitioners, and field roles without desktop or laptop access theyareoftentheo*nly*platformavailable.Theyarecarried duringwalk-throughs,physical engagements, andtravel-heavy assignments.Yettechnicaltestingmuststillrespects*cope*, *authorization*, and *traceability*. Cloud-centric platforms solve collaborationandcentralpolicyenforcementbutintroduce dependency,dataresidencyquestions,andoperationalfriction offline. *Local-first*toolchainsaddressconnectivityandcustody butriskunder-documentedtestingifgovernanceisleftentirelyimplicit.

AssessArc targets that setting without abandoning gover-nance: authorized analysts run reconnaissance scans, manual HTTP experimentation, encoding and analysis helpers, and local review of findings on the handset itself to identify vulnerabilities in permitted targets, all *without* a separate PC toolchainoravendor-hostedbackend.Thedesignquestion is how to structure such an application so that (a) high-risk networkemissionisprefacedbyexplicitin-appRoEstate,
(b)evidencetypesremaindistinguishable(formalscanexports vs. informal notes), and (c) operators retain fine control over sequencing and interpretation.

A. Design Principles

Fourprinciplesrecur:

1) *Human-in-the-loop control.* Scans start and stop under explicit user action; modules do not silently chain into one another. This reduces accidental amplification of traffic and keeps causal attribution with the operator.
2) *Governance where exposure is highest.* Active network reconnaissanceandtheprimaryHTTPlaboratoryrequire a recorded engagement before traffic is emitted. Lower-risk or offline transformations remain available under procedural policy.
3) *Separation of evidence types.* Flat-file scan exports, encrypted engagement context, and informal workflow notes follow distinct persistence paths, so reviewers can reason about evidential strength.
4) *Provenance without over-claiming.* In-app RoE fields and optional metadata footers support traceability butdo not replace legal contracts, change-control records,or external ticketing systems.

©IJRASET: All Rights are Reserved

B. Contributionsandpaperorganization

This paper is *methodological*: it explains what AssessArc does,howsubsystemsinteract,andwhichtrade-offsfollow,at a level suitable for peer review. Symbol-level implementation detail is intentionally secondary and may be supplied as supplementary material.
Section II situates the work. Section III deepens the RoE lifecycle. Section IV describes application composition. Sec-tions V–VIII develop VA, PT tools, utilities, and the report repository.SectionIXintegratescross-cuttingworkflows.Sec-tionXdiscusseslimitationsandethics.SectionXIconcludes.

II. RELATED WORK

Structured penetration-testing methodologies, including PTES phases, the OWASP Web Security Testing Guide (WSTG) categories, and NIST SP 800-115 technical exami-nationguidance,emphasizescoping,authorization,repeatable procedures, and traceable findings [1]–[3]. AssessArc maps naturallyonto*intelligencegathering*,*threatmodeling*,and*vul-nerability analysis* phases in such frameworks, while leaving exploitation and formal reporting to organizational process outside the app.
Mobile platforms impose *sandboxing*, permission models, andstorageisolationthatshapehowtoolspersistcreden-tials, findings, and network history [4]. Local assessment artifacts therefore inherit platform confidentiality guarantees (app-private storage) but also platform limitations (no built-in enterprise ACL on device, plaintext files unless additionally protected).
Priorworkon Mobile security tools pansmalware analysis, network monitors, and educational labs; fewer publications focuson*field*PT/VAworkbenchesthatforegroundRoE*before* reconnaissance. AssessArc contributes an explicit decompo-sition: encrypted RoE preferences, gated VA and primary HTTP lab, file-based VA evidence with deduplication, and SharedPreferences-basedPTworkflownotes;thissplitmirrors both *risk* and *evidentiary* roles.

III. RULES-OF-ENGAGEMENT LIFE CYCLE

A. Semanticdatamodel

Before high-risk operations, the application records a struc-tured snapshot intended to mirror a lightweight RoE packet:

- Organizational identity and authorized scope text (human-readable boundaries).
- Ticket or reference and named approver and assessor fields (accountability anchors).
- Avalidity window:durationfromactivation,expressedin hours and bounded to a configurable maximum, so that long-forgotten engagements do not indefinitely authorize traffic.

Operators create, update, or clear the record from the home surface; reopening the dialog pre-fills existing values to sup-port renewal or correction mid-engagement.

B. Confidentialityandfreshness

The record is protected using *encrypted application pref-erences* (platform-backed key derivation and authenticated encryption).Thismitigates*casual*disclosureifdevicebackups or rooted inspection expose preference XML, compared with cleartext key–value stores. It does *not* constitute a hardware security module or enterprise key escrow.
Temporal validity is enforced on read: expired records are treatedasabsentandcleared,reducingtheriskthatstalescope silentlyauthorizesnewsessions.Thisisa*conservative*default: falsenegatives(blockedscansuntilrenewal)arepreferredover false positives (unbounded reuse).

C. Selectiveenforcementrationale

All VA modules that perform *network reconnaissance* con-sult engagement state before running.

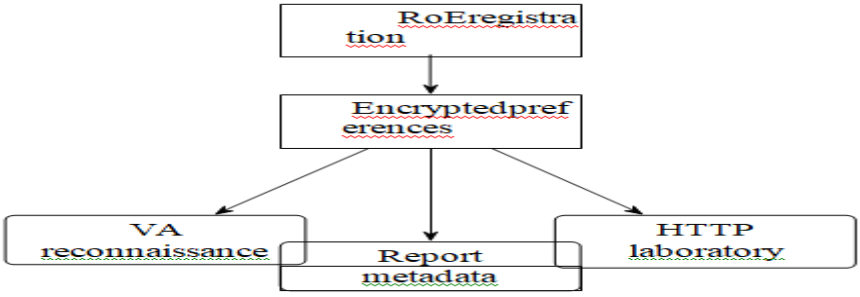


Fig. 1.Engagement registration, secure store, gated network features, andoptional report metadata.

The HTTP Request LaboratoryinPentestToolsappliesthesamecheckbefore transmittingrequests. Rationale:thesepathsgenerate attacker-observable traffic at scale or arbitrary HTTP toward user-supplied endpoints.
Conversely, offline helpers (encoding, hashing, JSON for-matting), paste-based static analyzers, and several auxiliary HTTP testers remain available without that *software* gate. Rationale:(i)benigntransformationsareoftenneededdur-ing analysis without implying target contact; (ii) some live tests are lower-volume or analyst-judgment-driven; (iii) over-gating can encourage workarounds outside the instrumented path. *Organizational policy* remains the ultimate control; the architecture documents where software enforcement is strict vs. procedural.

D. Couplingtosavedartifacts

When VA or SSL outputs are written to flat files, thesystem may append an *engagement metadata block* if an engagement exists at save time, echoing organization, scope, ticket, approver, assessor, and validity end. This strengthens *artifact provenance* when reports are exported or archived. It must not be mistaken for standalone legal authorization.

IV. APPLICATION COMPOSITION

The*home*surfaceroutestofiveareas:RoEregistration,util-ities (offline helpers), vulnerability assessment, pentest tools (tabbed hub), and saved reports. Navigation is *hierarchical*: each area exposes a hub (card list or tab strip) that leads to specialized screens. This favors *discoverability*, *incremental testability*ofmodules,and*lowcoupling*betweenfeatureteams or future contributors.
The VA and PT hubs differ superficially (cards vs. tabs)but share the same intent: aggregate many tools without embedding or chestrationlogicthatwouldcouplescanordering to code paths.

V. VULNERABILITY-ASSESSMENT METHODOLOGY

The VA suite comprises four modules as presented in the product interface: SSL Check, Directory Scan, Subdomain Scan, and VHost Scan. They address complementary recon-naissance questions: transport and surface hardening, content path discovery, hostname inventory expansion, and name-based virtual hosting on shared infrastructure.

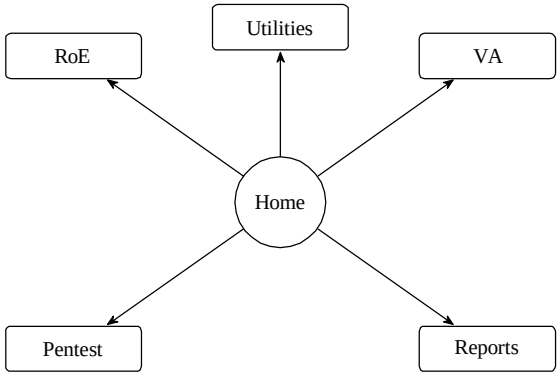


Fig.2.Primaryroutingfromthehomesurfacetomajorsubsystems.

A. Sharedplatformbehavior

Modules share a common *HTTP client layer* with config-urable timeouts and redirect policy. *No-redirect* clients are usedwhereautomaticfollowingwouldmaskstatuscodes(e.g., redirect classification and brute-force style path probes). A consistentuser-agentstringreducesfingerprintvarianceacross modules.
Input normalization validates URLs and domains, trims wordlists,andcomposescandidatepathsorhostnames.*Up-per bounds* on candidate counts reduce accidental denial-of-service against targets or local resource exhaustion.
Long-running work executes off the UI thread; operators may *stop* iterative scans atomically, preserving partial results and progress messaging.

B. SSLCheck

SSL Check evaluates a primary URL in layers: (1) for non-HTTPS entry URLs, an HTTP GET *without* following redi-rects inspects status and Locationto infer whether trafficis steered toward TLS;

(2) a TLS handshake yields certificate chainandnot-afterinformation;(3)selectedsecurity-sensitive response headers are collected; (4) Set-Cookieattributes are parsed for common hardening attributes. A *risk score* aggregatesweightedsignalsfortriage.Persistenceis*operator-initiated*: the analyst saves a snapshot after review, serializing the layered results into a textual report.

C. SubdomainScan

Given a validated root domain and a label wordlist, the tool formscandidates⟨*label*⟩.⟨*root*⟩.Foreachcandidateitperforms *DNS resolution*; on success it may probe https://andthenhttp://roots to obtain application-layer status codes. A short inter-request delay may throttle burstiness. Results aggregate hostname, resolved address, and best-effort HTTP status,supportingmanualprioritizationforfollow-ontesting.

D. DirectoryScan

Given a base URL and a path/wordlist (plus optional file-nameextensions),thetoolcomposescandidateURLs,requests each, and classifies responses to surface *reachable* paths. Candidatevolumeiscappedbeforeexecution.Interpretation remainsheuristic :statuscodesandlengthsindicate*candidates for review*, not definitive vulnerability.

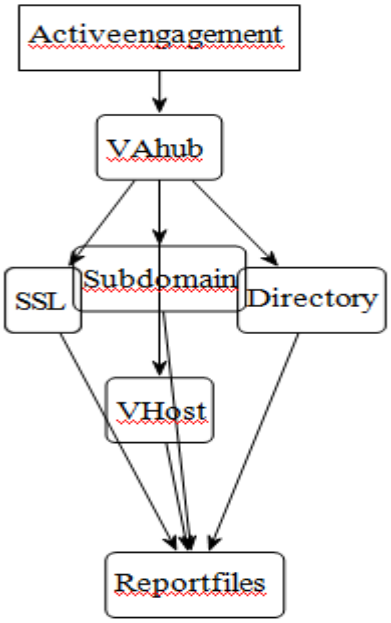


Fig.3.VAhub,modules,engagementprecondition,andflat-fileoutputs.

E. VHostScan

Given a fixed base URL (often an IP or shared endpoint) and host-header candidates (optionally suffixed with a root domain), the tool establishes a *baseline* response using a control hostname, then compares each candidate's status and body length (and related signals) to flag *divergences* sug-gestive of distinct virtual hosts. This supports shared-hosting reconnaissance; false positives arise from caching, CDNs, and dynamic content.

F. Evidencepolicy

SSL Check writes reports on explicit save. The three itera-tivescannerswriteflatfileswhenthein-memoryfindinglistis non-emptyatstoporcompletion;emptyrunsskipfilecreation to avoid noise.

A recommended reconnaissance ordering (illustrative only) is: SSL Check →Subdomain Scan →Directory Scan → VHost Scan. The software does not enforce ordering; analysts may iterate or repeat modules as methodology dictates.PENTEST-

VI. PENTEST-TOOLS METHODOLOGY

Pentest Tools organizes twelve capabilities into four tabs (Core, Analysis, Helpers, and Workflow), as summarized in Table I.

A. *Networkvs.localtools*

Manytoolsoperateonpastedortypeddatawithoutnetwork I/O, supporting *air-gapped analysis* of captured traffic. Live HTTP tools share client conventions with VA. As stated in Section III-C,*only* the HTTP Request Lab requires an active engagement before sending; other live testers rely on operator judgment unless policy extends gates.

B. *Workflowpersistencesemantics*

Workflow notes and checklists persist in *lightweight local preferences*, distinct from flat-file VA reports. This supports session continuity and personal methodology tracking but is *not*presentedasatamper-evidentormulti-userevidencevault.

TABLEI
PENTESTTOOLSTABSANDCAPABILITIES(PRODUCTLABELS).

Tab	Capabilitiesandmethodologicalrole
Core	HTTP Request Lab: arbitrary method, URL, headers, and body with timing and full response capture (engagement-gated beforesend). Encoder: ordered encode/decode chains (URL, Base64, hex, Unicode, HTML) with optional double pass for chainedobfuscation studies. JWT Inspector: Base64url decoding of header and payload [5], calendar expiry interpretation, and heuristic“risk flags” around algorithms and claims (analyst must validate). Diff Viewer: line-oriented comparison in text, JSON, or headermodes to summarize additions, removals, and edits after mutations.
Analysis	Response Analyzer: static analysis of pasted response headers/body (and optional probe string) for CSP, CORS, cache, cookieattributes, and reflection hints; intended as triage, not exhaustive policy proving. Auth Header Tester: replays a target URL withBearer, Basic, and API-key style variants to compare behavior. CORS Tester: issues crafted cross-origin requests to observebrowser-relevant policy behavior for configured origins.
Helpers	Wordlist Mutator:generatesprefix/suffix/case/numericpermutationsfordownstreammanualuse.SSRF URL Helper:composescandidate URLs for *authorized* SSRF hypothesis testing. Open Redirect Tester: builds redirect-parameter payloads from a baseURL and parameter name for manual validation.
Workflow	Scope Notes:free-formtarget/endpoint/auth/finding-drafttext.IDOR ChecklistandJWT Attack Checklist:structuredchecklistswith template load/save for repeatable methodology tracking.

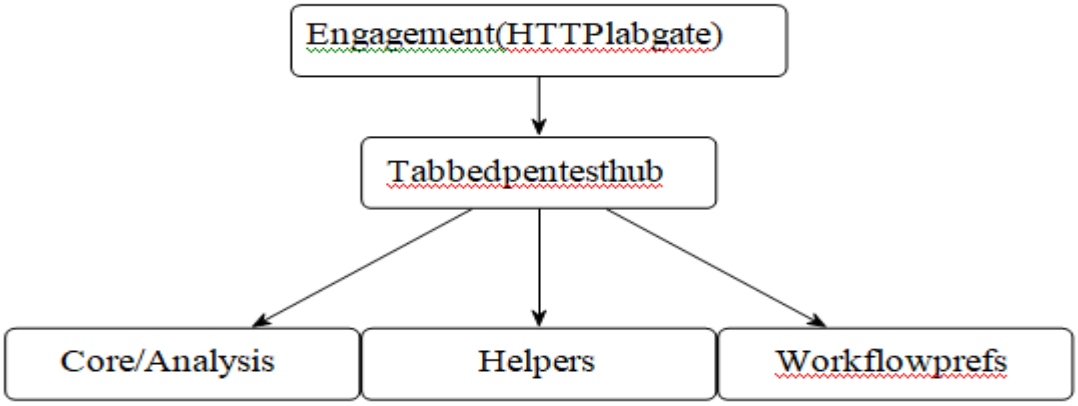


Fig.4.Pentesthub:engagementgateontheHTTPlaboratory;workflowstatein separate storage.

VII. UTILITIES METHODOLOGY

Utilities exposes three cards in the shipped configuration: Encoder (the same transformation engine as in Pentest Tools, ensuring consistent semantics across entry points), Hash tool (hexadecimal digests over UTF-8 input with selectable algo-rithms from a configured set), and JSON tool (pretty-printing and syntactic validation). Paths are predominantly *offline* and not engagement-gated in software, reflecting a design bet that benign transforms should remain available while misuse is governed by policy.

Cross-area composition is *manual*: encoded payloads or formatted JSON are copied or retyped into VA/PT surfaces, preserving explicit analyst intent at each hop.

VIII. SAVED REPORTS METHODOLOGY

Saved Reports is the *reader* for flat textual outputs from SSL Check and the three iterative VA scanners.

A. *Repositorybehavior*

Files are listed in reverse chronological order by last modification. Substring search filters filenames. Each row expandstoshowfullbodytextondemand.Theplatforms*share* intentexportsplaintexttootherapplications(email,notebooks, ticketing)underusercontrol.Bulkdeleterequiresconfirmation andremoves report files and associated deduplication indexes.

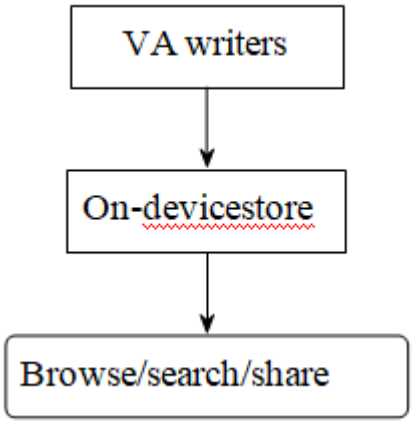


Fig.5.Producers,localtextualstore,andreviewUIforsavedreports.

B. *Deduplicationandintegritylabeling*

Each save constructs a typed header (module class, times-tamps, target context, progress), body lines, optional engage-ment footer, and a *content fingerprint* line derived from SHA-256 over a *canonicalized* representation that excludes volatile timestamp and signature lines. Before writing, the store com-pares against existing material so that *logically duplicate* reports need not multiply files, which supports repeated runs without clutter while preserving distinct runs when content differs.

Pentestworkflownotesdo*not*populatethisrepository,rein-forcingtheseparationbetween*scanevidence*and*methodology scratchpad* data.

IX. CROSS-CUTTING WORKFLOWANDINTEGRATION

A coherent field session may proceed as: register or refresh engagement; run VA reconnaissance and save reports; review exportsinSavedReports;usePentestToolsfortargetedtesting (HTTP laboratory respecting engagement); use Utilities for encoding and JSON checks; share artifacts under policy; clear local reports or engagement on handset reassignment.

A. *Manualcompositionasadesigntrade-off*

Data do not traverse a single internal pipeline object. Operators bridge tools by copying, retyping, or rereading outputs. This trades *automation efficiency* for *transparency*: eachemissionortransformationstepremainsvisibleand deliberate,whichaidstraining,court-defensiblenotes,and scope discipline at the cost of integration labor.

B. *Positioning*

AssessArc is best described as a modular workbench: one home composition, heterogeneous persistence models, one cross-cutting engagement concept for the riskiest network features, and file-based evidence for reconnaissance-heavy outputs, with field adaptability preferred over end-to-end or-chestration.

X. LIMITATIONS AND ETHICAL CONSIDERATIONS

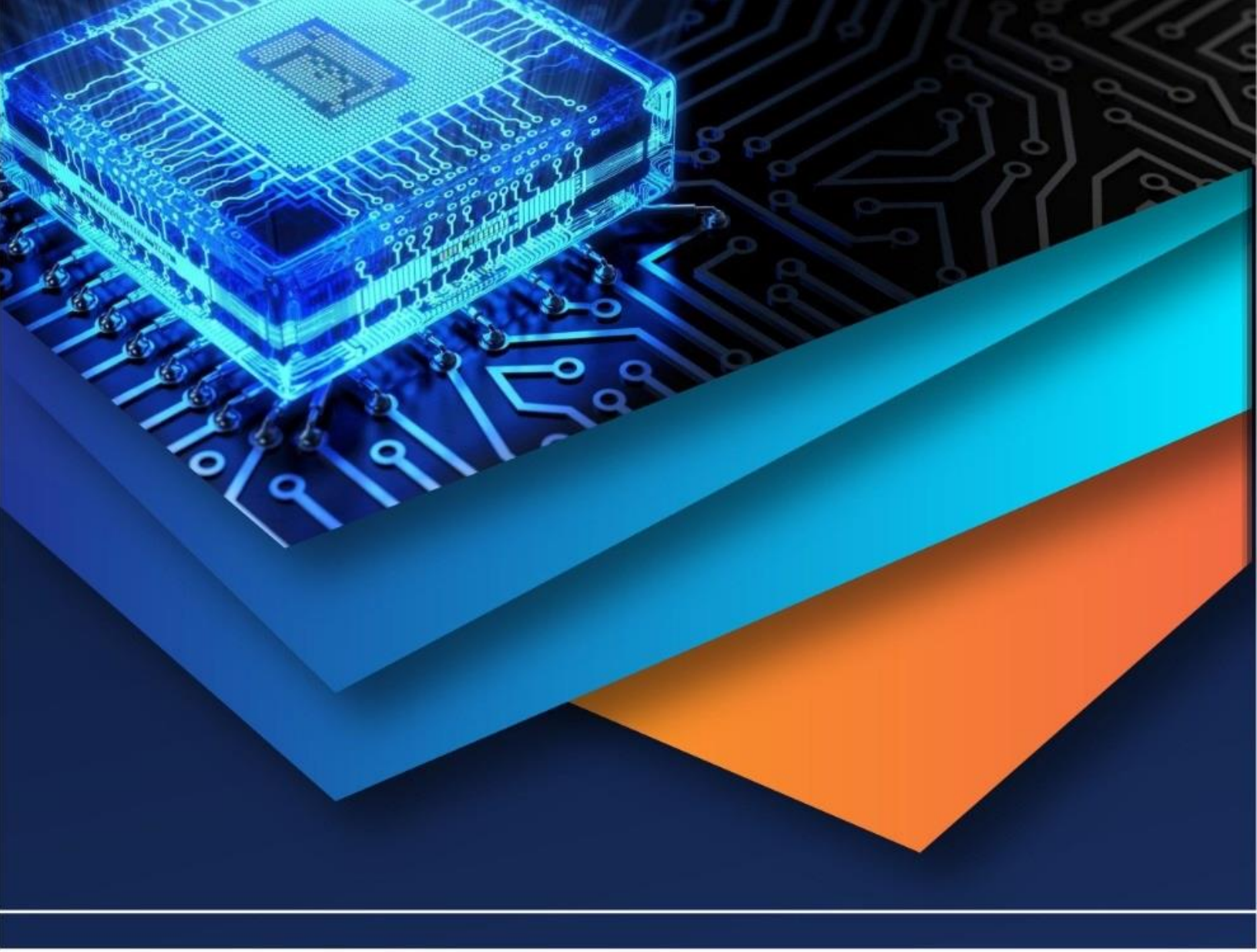
1) *Governance:* Engagement fields are self-asserted; there is no built-in verification against enterprise ticketing or identityproviders.Enforcementispartial(SectionIII-C).Only one active engagement is modeled.
2) *Technical accuracy:* Reconnaissance interpretation depends on heuristics (status codes, lengths, headers); false positives and negatives require human validation. Network results may be influenced by middleboxes, CDNs, and rate limiting.
3) *Storage security:*Report files are local and plaintext within the app sandbox; there is no cloud sync or on-device role-based access control. Encrypted preferences protect RoE relativetocleartextsettingsbut notallthreatmodels(e.g.,full device compromise).
4) *Ethics and law:* Researchers and practitioners must use such tools only on systems they own or are explicitly authorized to test, in compliance with law and organizational policy. The tool enables assessment; it does not substitute for authorization.

XI. CONCLUSION

We described AssessArc as a mobile, modular workbench coupling RoE recording, VA reconnaissance, pentest aids, utilities, and local report review, with deep methodological patterns for each surface. The architecture emphasizes selec-tive gating, heterogeneous persistence, deduplicated textual evidence,andmanualcompositionbetweentools.Futurework may explore external ticket integration, stronger evidence packaging(e.g.,signedexports),formalmodelsofengagement state, and broader automation that stays within policy.

REFERENCES

[1] Penetration Testing Execution Standard (PTES) Technical Guidelines.http://www.pentest-standard.org (accessed 2026).
[2] OWASPFoundation,“OWASPWebSecurityTestingGuide,”stablereleasedocumentation.https://owasp.org/www-project-web-security-testing-guide/ (accessed 2026).
[3] NIST, “Technical Guide to Information Security Testing and Assess-ment,” SP 800-115 Rev. 1, 2020.
[4] Google, “Android Security Overview,” Android Open Source Projectdocumentation (app sandbox, storage model).
[5] M. Jones, J. Bradley, and N. Sakimura, “JSON Web Token (JWT),”IETF RFC 7519, May 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)