APPENDIX-A

Request for Proposal (RFP)

for

Platform for Android OS Integrity & Patching (PAOIP)

Scientific Analysis Group (SAG),
Defence Research & Development Organization (DRDO)
Ministry of Defence, Government of India

Contact Details:
Director, SAG, Metcalfe House Complex, Delhi-110054, India
Tel: +91-11-23819828, Fax: +91-11-23812683

TABLE OF CONTENTS

	Content	Page No.
Section 1	General Information and Purpose	4
1.1	Scope and Overview	4
Section 2	Functional Requirement for PAOIP	5
2.1	Conceptual Block Diagram of PAOIP	5
	2.1.1 Subsystem 1: Android Platform Integrity	7
	2.1.2 Subsystem 2 : Patch Security Analysis	8
	2.1.3 Subsystem 3 : Integrity Impact Analysis	8
	2.1.4 Subsystem 4 : Central Manager	9
2.2	Technical Specifications	11
Section 3	Non-Functional Requirement of PAOIP	16
3.1	Design Framework	16
3.2	Time Frame for Delivery	16
3.3	Venue for Carrying Out Work	16
3.4	Modularity	16
3.5	Graphical User Interface	17
3.6	Design Consideration for Performance Optimization	17
	3.6.1 Incident Logging	17
	3.6.2 Configurability	17
3.7	Documentation	17
3.8	Training	18
3.9	Vendor Qualification Criteria	18

3.10 Mode of Selection 18

3.11 Acceptance Testing 19

3.12 Non-Disclosure Agreement (NDA) 19

3.13 Warranty/Maintenance 19

3.14 Transfer of Source Code 19

3.15 Review, Audit and Termination 20

3.16 Deliverables 20

3.17 Ownership of Intellectual Property (IP) 21

Section 4: GANTT CHART 23

Section 5: Technical Compliance Report 24

Section 6: Project Resource 35

Cost Estimation 36

Platform for Android OS Integrity and Patching

Section 1: General Information and Purpose

Scientific Analysis Group (SAG), a premier lab of DRDO invite a proposal for supply of ***Platform for Android OS Integrity and Patching (PAOIP)*** which aims at development of customized solutions comprising of patch security analysis for known vulnerabilities, methodology for ensuring integrity of Android platform and solution for patch impact analysis. For the given test patch, solution should deduce inference from patch security analysis and integrity impact analysis about its acceptability and effect on platform. In addition, platform should have provisions for user configuration manager and result visualization.

The contents of this Request for Proposal (RFP) are strictly meant to develop a custom-built solution for SAG against this tender enquiry and they are to be treated in confidence and are not to be revealed directly or indirectly to any entity not concerned with the proposal.

It is required that the respondents to this RFP should perform an exclusive transfer of complete source code, design documentation, user document, AI/ML models and related training data, associated hardware platform and details of the know-how.

1.1 Scope and Overview

The scope of the work involves supply of Android OS integrity and patching platform comprising of hardware and software modules to be given by the solution provider. The Android devices under scope may belong to manufacturers who uses stock Android, or share the code base of their customized Android OS and drivers in the public domain (Open source kernel code).

PAOIP mainly consist of four subsystems namely *Subsystem 1*: Android Platform Integrity; *Subsystem 2:* Patch Security Analysis; *Subsystem 3:* Integrity Impact Analysis; *Subsystem 4:* Central Manager. The following are the major modules to be developed for the realization of PAOIP.

- AI/ML Models
- Inference Models

- Optimized Mechanism for Android Platform Integrity
- Static patch Security Analysis
- Dynamic patch Security Analysis
- Analysis & Decision Model
- Static Impact Analysis
- Dynamic Impact Analysis
- User Interface/ Configuration Manager
- Reporting and Logging
- Synthetic Patch Data Generator
- Patch Database

The customized solution for Platform for Android OS Integrity and Patching should fulfil all functional and non-functional requirements, with minimum hardware details as mentioned in this RFP. This document describes functional and non-functional requirements of *Platform for Android OS Integrity and Patching.*

Section 2: Functional Requirement for PAOIP

Functional requirements for Platform for Android OS Integrity and Patching are illustrated through conceptual block diagram along with technical specifications. The details are given in thereafter.

2.1 Conceptual Block Diagram of PAOIP

PAOIP consists of four major subsystems, namely *Subsystem 1*: Android Platform Integrity; *Subsystem 2:* Patch Security Analysis; *Subsystem 3:* Integrity Impact Analysis and *Subsystem 4:* Central Manager; each having its own set of functionalities. These subsystems consists of various modules that work in cohesive manner to ensure not only faithful patching of Android device but also gauge the effect of patching on the integrity of device platform. Conceptual block diagram of **PAOIP** highlighting the major modules along with their intended functionality is shown in **Figure 1** below.

Android Platform Integrity methodology needs to be devised by the solution provider in order to ensure the minimum effect of new test patch on the established integrity mechanism of Android platform. For this, patch database of different version of Android OS along with synthetic data generator may be considered to train the AI/ML model for devising integrity mechanism. Design and development of optimum mechanism for Android platform integrity is absolute requirement and work as input for subsystem 3 wherein effect of new patch on established integrity of Android platform is to be determined.

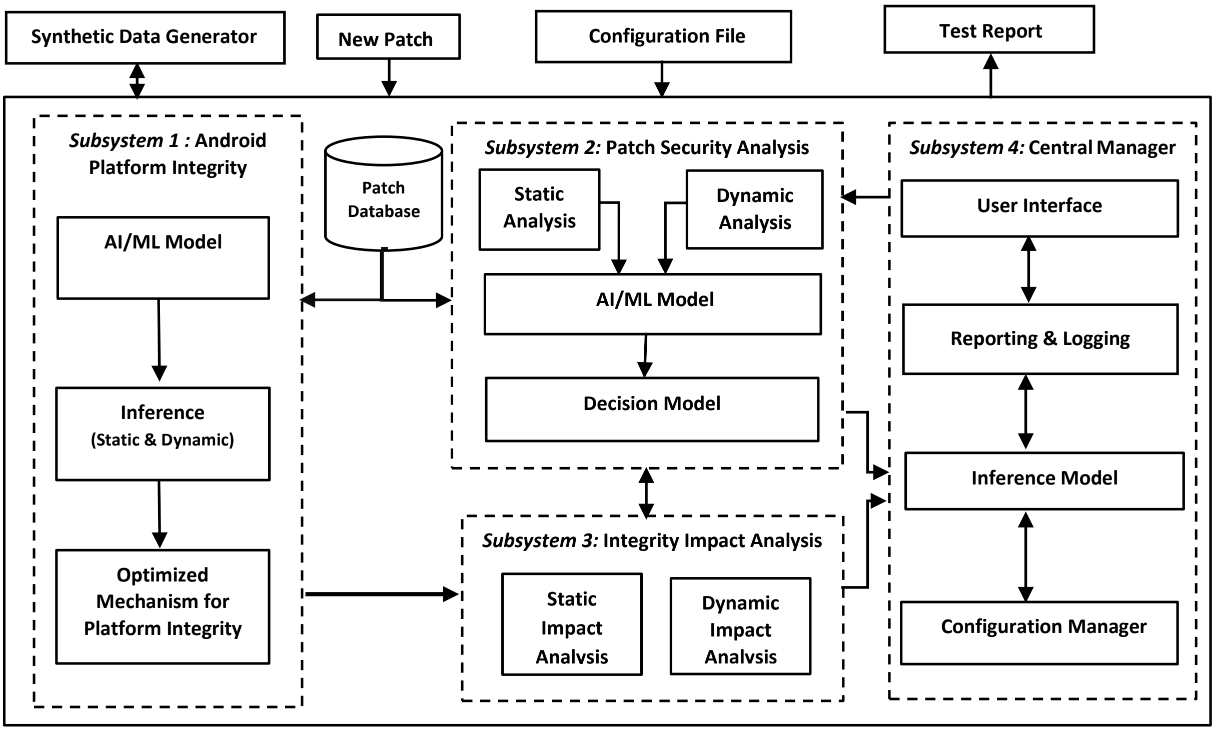

Figure 1: Conceptual Block Diagram

Patch Security Analysis should be able to perform static and dynamic security analysis of new patch. For this, AI/ML model should be sufficiently trained with the different variant of patches for different version of Android OS. Effective decision model needs to be devised and results to be communicated to Integrity impact analysis & control manager subsystems.

Integrity Impact Analysis should be able to gauge the impact of patch on the establish integrity of Android platform. For this, both static and dynamic impact of given patch on the integrity of platform is determined and results are communicated to central manager for deducing inferences.

Central Manager should not only be able to control and monitor the patch security analysis and integrity impact analysis but also be able to deduce inference about a given test patch for the Android device. It should also be able to control & update the patch database, patch security analysis and Integrity impact analysis for various user parameters/configurations for effective analysis. For this, GUI should be user friendly to enable input parameters/configurations and also effectively visualize the analysis results in specific format.

Modules needs to be implemented on specific hardware with sufficient RAM and storage for effective analysis and decision. The details of all four subsystems of PAOIP along with brief description of their individual modules follows in turn.

2.1.1. Subsystem 1: Android Platform Integrity - This subsystem aims at development of methodolgy for Android platform integrity such that effect of new test patch on estiblished integrity mechanism is minimized. Android Platform Integrity mainly consists of software modules namely AI/ML model, Inference (Static and Dynamic) and optimized mechanism for platform integrity. The brief descriptions for these modules along with their intended functionality are given below:

a) *AI/ML & Inference Model:* These modules would involve the creation of an AI/ML model based on the past security patches for finding the parts of the Android platform which are more and less affected by these patches. For this, modules need to establish correlation between Applications/Libraries/Drivers/Interfaces/Kernel etc. with the security patches. Solution provider needs to create the database of latest patches from Android 9 onwards from publicly available databases like CVE. These patches specifically need to be security patches with high security critical patches given the most priority. The effective features such as severity level of the patches, patch information, android version, sensor / transducer affected for which patch is released, timeline etc. may be deduced for training the AI/ML model. Inference model should partition the Android platform into two sets: Static Partition and Variable Partition. Here, static partitions are those which are rarely affected by patches and variable partitions are those which get frequently affected by the patches. Moreover, the binaries may be divided into separate partitions along the lines of Android OS stack such as Boot partition, Kernel partition, Framework/System/HAL partition and application partition. The results of these modules will be used for establishing an optimum Integrity mechanism for Smart Devices.

b) *Optimized Mechanism for Platform Integrity:* This module uses the results of previous modules to devise the effective & optimum integrity mechanism of Android platform. Aim is to ensure data integrity of Android platform running on the Smart Device. In this, for instance, integrity mechanisms may be considered as Hash/Hash chain, Public Key Cryptography for secure booting of smart device platform. Established integrity mechanism should be implemented on the sample test devices to be supplied by the solution provider.

2.1.2. Subsystem 2 : Patch Security Analysis -This subsystem aims to validate the given test patch for the known vulnerablities/malwares for the purpose of security analysis. This subsystem should have provision to update for new vulnerabilties/malwares. Patch Security Analysis mainly consists of static & dynamic analysis, AI/ML and Decision Models. The brief descriptions for these modules along with their intended functionality are given below:

a) Static and Dynamic Analysis: The primary objective of this module is to perform static and dynamic analysis of given test patch. Solution provider would devise effective features for the patch and accordingly train the AI/ML model while considering open source databases and synthetic dataset. Apart from this, during static analysis, compliance w.r.t. security coding standards such as OWASP, CWE, CERT-In etc. may be checked. Many other aspects such as source of patch, severity level of vulnerabilities etc. may be considered for static analysis. For dynamic analysis, the source code of Android platform is updated with the patch and accordingly binaries are build. These updated binaries need to be flashed onto a test device or run in an emulator. This setup needs to analyse & capture change in the behaviour & functionality of the device and also malicious activities etc. during the dynamic analysis.

b) AI/ML & Decision Models: These modules should mainly process the inputs from static and dynamic analysis modules to deduce inference and decision about acceptability of test patch. These models should mainly detect/Infer the malicious activities/abnormal behaviours and untoward incidence. For malware detection, model should be trained with sufficient number of benign applications and latest malicious datasets. In this, for instance, Database such as Android Malware Genome Project, Drebin Dataset, AMD Project, AAGM Dataset etc. may be considered for training the AI/ML model. Rule based decision model may also be considered for effective decision modelling about acceptability of test patch for patching the Android platform.

2.1.3. Subsystem 3 : Integrity Impact Analysis- This subsystem aims at development of toolkit for gauging the impact of patch on the Android platform integirty. For this, module takes the binaries/source code of Android platfrom, know how of its integrity mechanism and valid test patch as inputs and deduce the impact of patch on the Android platform integrity. Integrity impact analysis mainly consists of static impact analysis & dynamic impact analysis. The brief descriptions for these modules along with their intended functionality is as follows.

(a) Static Impact Analysis & Dynamic impact Analysis: These module would cover the Post Patch Impact Analysis. If the patch is benign in nature as per the analysis in the subsystem 2, then impact analysis of the patch on the Smart device needs to be performed. It will have two components, one is the analysis at the file level/ partition level (kernel, system etc.) changes in the Android platform and other is the analysis of its effect on different applications/libraries/system calls/ inter process communication etc. The result of this analysis will be used by the *Inference Model* which will decide whether this patch leads to minimal/huge changes in integrity and functionality of the Smart Devices along with security risk (low/medium/high). This will also give hot map for Android platform after new patch execution vis-à-vis old version to ascertain the effect of patching. This will help in gauging the effect of patch execution to aid Inference model in subsystem 4.

2.1.4. Subsystem 4 : Central Manager- Central Manager for PAOIP comprises of software modules which mainly consists of user interface, reporting and logging, inference and configuration manager. The brief description for these modules along with their intended functionality is given below:

(a) User Interface & Configuration Manager: These modules should able to control & update the patch database, patch security analysis and Integrity impact analysis for various user parameters/configurations. Modules should allow user to choose the Security Patch, Android OS version and device on which the patch needs to be applied along with the integrity mechanism for enabling the secure boot of the Android platform. The module should have default configuration for system and allow to change to any other configuration based on user requirements. The system may adapt to the new configuration set by user and should start operating based on the new configuration parameters. In sum, the module will provide a mechanism to configure parameter for different test options, input data types, log and save option, report format defined/selected by user.

(b) Inference Model: This module can have AI/ML/Rule based analysis and decision system which will have capability to analyse the reports and results of subsystem 2

and subsystem 3 and infer/deduce informed decisions about the patched Smart device. In other words, this module should consider various parameters like security level of the patch, effect on critical applications/libraries, overall impact of patch on the integrity of the smart device etc. for inferring about patching of smart device. This model should give a cumulative score based on its findings and report this to the reporting and logging module for summarization of result.

(c) Reporting & Logging: This module should generate the output report in the specified formats along with visualization of results in chosen or selected formats. Moreover, it should also support logging of the data in three formats - error log, warning logs & information messages; and events for different patches, such that, at any point user can enable either of these logs individually or all three for the purpose of analysis.

2.2 Technical Specifications

1. Solution provider should develop solution for Android platform integrity such that effect of new test patch on established integrity mechanism is minimized.

2. Solution should provide options for the user to select a version of Android source code and targeted device configuration.

3. Solution provider should provide development environment to build the AOSP code which needs to be downloaded from the official website of Google.

4. Solution should have the sufficient database of latest patches from Android 9 onwards from publicly available databases like CVE, official Google website etc. These patches specifically needs to be security patches with high severity level patches given the most priority. Also, provision for updating patch database through GUI to be provided.

5. Solution should have trained AI/ML model based on the past security patches or synthetic test data generator. The effectiveness of extracted features from patch database needs to be ensured for achieving performance benchmarking (F1 score, Accuracy, Precision).

6. Inference model should partition the Android platform into two sets: Static Partition and Variable Partition. Solution should also ensure to separate the binaries into separate partitions along the lines of Android OS stack such as Boot partition, Kernel partition, Framework/System/HAL partition and application partition.

7. Solution provider should develop and implement mechanism for Android platform integrity based Hash/Hash chain, Public Key Cryptography for secure booting of smart device platform. Solution should have cryptographic Safe Hashes for development of Hash based integrity Mechanism.

8. Testing and validation of integrity mechanism should be performed on two Smart Devices (preferable Samsung and Google Nexus Series devices) which needs to supplied by the solution provider.

9. Solution should store the golden hashes of the different partitions of the Android OS in a secure location with restricted user access rights.

10. Solution should able to perform static analysis of given test patch for the known vulnerablities/malwares for the purpose of security analysis.

11. Solution should able to perform dynamic analysis of given test patch after either by applying to the test device or by running it in the emulator.

12. Dynamic analysis should able to detect and report difference in behaviour/functioning between Patched Smart Device and Unpatched Smart Device. Any unexpected change needs to be logged and highlighted with summary.

13. Solution should be able to process the inputs from static and dynamic analysis modules to deduce inference and decision about acceptability of test patch. It should be able to detect/Infer the malicious activities/abnormal behaviours and untoward incidence if any for the given test patch.

14. Solution should have provision to gauge the impact (static impact analysis and dynamic impact analysis) of patch on the established Android platform integrity.

15. Soulution should able to perform impact analysis at two levels namely one is the file level/partition level (kernel, system etc.) changes in the Android platform and other is the analysis of its effect on different applications/libraries/system calls/ inter process communication.

16. Solution should be able to process the inputs from static and dynamic impact analysis modules to deduce inference and decision about whether patch leads to minimal/moderate/huge changes in integrity and functionality of the smart devices along with level of associated security risk.

17. Solution should have User Interface to allow user to choose the Security Patch, Android OS version on which the patch needs to be applied along with the integrity mechanism for enabling the secure boot of the Android platform.

18. Solution should be able to perform AI/ML/Rule based analysis and decision for the results obtained through Android platform integrity and patch security analysis module.

19. Solution should be able to generate cumulative score about overall impact of patch on the integrity of the smart device considering parameters as security level of the patch, effect on critical applications/libraries, overall impact of patch on the integrity of the smart device.

20. Solution should be able to logged all the errors, results, exceptions etc. using Logging Module. It should support logging of the data in three formats - error log, warning logs & information messages.

21. UI Module/ Control & Configuration Manager should provide a provision to configure parameter for different test options, input data types, log and save option, report format defined/selected by user.

22. Solution should be able to take user configuration data either as *.xml, *.csv, *.json file or input from GUI and reconfigure.

23. Hardware should be from standard OEM provider with warranty and logo emboss to the product/hardware.

24. GUI of the solution should be simple, smooth and interactive with help guide for easy handling of the solution by the user.

25. Solution should work on rooted devices of the manufacturer who use stock Android, or share the code base of their customized Android OS and drivers in the public domain (Open Source Kernel Code) with root access.

26. Solution should achieve high precision & F1 Score value for AI/ML module which is optimally trained with sufficient data.

27. Solution should achieve minimum accuracy of 90% for different defined inferential parameters for implemented AI/ML module.

28. Solution should have provision to train the AI/ML module with new user defined class of malware by invoking input data either from patch database or from test data generator.

29. Solution should have provisioning for generating the combined analysis results in *.pdf and Microsoft Word format.

30. Solution should have provision for operator to enable individual logs or all logs for the purpose of diagnosis & Information purpose.

31. Solution should provide features to manage users (add new users to the system, edit user details, delete users, change their permissions).

32. Solution should support and implemented on two test devices (preferably Samsung S Series and Google Pixel).Test device to be supplied by the solution provider.

33. Solution should handle any unexpected errors or exceptions within the concerned module, without causing any system instability. Errors and exceptions should be logged for further review.

34. Solution should execute each module independently and each module should provide output only via clearly defined interfaces. Each module should implement input sanitisation methods such that only expected values are received from other modules.

35. Solution should be configured to be run in a containerised environment. The containers should be configured to handle concurrent processes and should be utilising the hardware resources optimally.

36. Solution should integrate the software components seamlessly.

37. Solution should utilise stable and maintainable technologies that are widely used for the customisable code base.

38. Solution should be developed using only buyer approved Free and Open Source technologies, libraries and software frameworks.

39. Solution should be a Bundled solution consisting of both hardware and software; solution provider should share the software component details in their application making their understanding of the platform clear.

40. Solution should have minimum hardware configuration of 64 GB DDR4 RAM & Intel Xeon Processor with form factor of 2U Rack.

41. Solution should have at least 1 TB SSD capacities

42. Solution should have a 2 TB storage HDD drive or higher.

43. Solution should have minimum 6 GB NVidia Graphic card.

44. Solution should have FHD Monitor of minimum 27″.

45. Solution should have a Dual power supply with a minimum 1100w fully redundant.

46. Solution should have USB, HDMI, and VGA ports.

47. Solution should be BIOS protected.

48. All the specified hardware should be from standard OEM.

Section 3: Non-Functional Requirement of PAOIP

3.1 Design Framework

Solution may be developed preferably in Python/Golang/Java/Kotlin/C/C++ with UI may be developed on React JS or Angular JS.

No proprietary software utility library should be used in the PAOIP. Software coding guidelines should be as per standards and should be explained in the proposal with example. Software coding guidelines should include commenting style, indentation, maximum line length to be used, line breaks, blank line, import, naming conventions and other standard practices being applied to the programming languages.

3.2 Time Frame for Delivery

The total duration of activity for the above task is 12 months from the date of placement of supply order which includes training as well as ATP. The timely completion of milestones and compliance is essential.

3.3 Venue for Carrying Out Work

The work will be executed by solution provider at his own premises with all resources like manpower, hardware platforms and requisite IDEs being the sole responsibility of the solution provider. Subsequent to completion of work, an ATP should be setup on hardware for compiling the source code to build and test the software with hardware for final acceptance.

3.4 Modularity

- Should ensure high cohesion among the modules.
- Each module should have clearly defined input and output interfaces.
- The output of a module can either be final output or used as input for other modules.
- The modules may be independently usable as a library or through APIs on suitable platform.
- The modular decomposition should be in line with decomposition of functional requirements of PAOIP as indicated above.

3.5 Graphical User Interface

A feature rich user friendly Graphical Interface to access smoothly the functionalities of the PAOIP is paramount. It should cater to user input requirements for effective usage of system.

3.6 Design Consideration for Performance Optimization

3.6.1 Incident Logging

a) System should log incidences like

  i. Successful execution of Patch security analysis.
  ii. Successful execution of Integrity Impact Analysis.
  iii. Successful creation of data which includes the analysis reports, analysis run history, registered devices, malware signatures.

b) Output of each module should be logged and reported with error messages in case of failure in execution.

c) Reports on patch, processed and stored should be generated for DUT subjected for PAOIP.

3.6.2 Configurability

PAOIP should have configuration options via configuration file for specifying device & Android OS version with patch selected, analysis to be done, event, logs (enable/disable) for system operation. Also, configuration can be specified through GUI/Command Line. A default configuration may also be given for the system to run in default mode.

3.7 Documentation

Bidder will provide extensive technical document, covering s/w design architecture, supported AOSP versions, inputs, outputs, features and operations of platform. All documents and artefacts pertaining to the S/w lifecycle should be maintained in preferably DSSD (DRDO Standard for Software Document) standards and delivered. This includes Software Requirement Specification, Interface Specification, Requirement Traceability Matrix, Software architecture document, Design Document, source code with description, test case plan, test report, user manual and installation manual.

3.8 Training

Solution provider will include an offer for at least one week long training for minimum 5 DRDO personnel at SAG Delhi/company premises. Training shall be provided on installation and usage of the software along with procedure for building and packaging the same. The training will cover the operation, maintenance and up gradation aspects of platform. Also, training should include code walk through and its mapping to design and requirements. In the offer, details of training shall be clearly enumerated. Training documents should be made available a week before the commencement of training.

3.9 Vendor Qualification Criteria

All Indian private and Indian public sector companies are eligible to take up this project. Companies should be in the domain of software development and testing. It is required that companies should provide query resolution (during the proposal evaluation) within 24 hours and that too in person visiting the SAG, DRDO office to clarify the queries or present supported documents. Failure to this may lead to disqualification of bidder. Bidder has to present the technical approach for the project covering all aspects and features with details of implementation approach, work flows and use case. Proof of concept (PoC) demonstration – Bidder has to give a demo of the basic level working PoC with a sample use case within the given time frame to the constituted technical expert committee. This is important evaluation criteria for the project award.

3.10 Mode of Selection

The competitive technical bids submitted by various bidders will be evaluated by a duly constituted TCEC committee for this purpose and the bidders technically found suitable and acceptable to undertake the work under this tender enquiry will be short listed.

The evaluation will include the following:

The respondents is required to give a technical presentation along with proof of concept and write-up on design consideration (with rationale) vis-a-vis the functional and non-functional requirements to the constituted technical expert committee (PoC Committee). The familiarity of the respondents with development tools and libraries required for the PAOIP and similar task executed in the past may also be highlighted.

The committee may seek additional clarifications on the technical bids and the approach paper for which a written response should be provided. Apart from compliance to the requirements, POC & clarity in approach will be key factors for short listing of prospective bidder.

3.11 Acceptance Testing

This PAOIP will be tested with respect to ATP (Acceptance Test Plan) criteria define during project scoping. Acceptance criteria will be broadly based on PAOIP performance (accuracy, precision, F1 score) in the detection of malicious activities/abnormal behaviours and untoward incidence if any for the given test patch. However, it will also cover Android Integrity Mechanism, patch impact analysis, UI acceptance for configuration, logging, user interface and dash board features.

3.12 Non-Disclosure Agreement (NDA)

Selected Vendor should be willing to sign a non-disclosure agreement to protect the contents of the data and the output results from various tests before release of supply order.

3.13 Warranty/Maintenance

Proposal should include for warranty and maintenance of the PAOIP for one year from project end date. During the maintenance, solution provider may be asked to support for installation and smaller modification. However, this maintenance support does not include any major change or feature enhancement implementation. Solution provider must cover for hardware platform warranty for 3 years in their proposal. After project completion, “Solution provider” shall return to the DRDO all classified Information belonging to the DRDO. In addition, and without limiting the generality of the aforesaid, the “Solution provider” shall return to DRDO, in their entirety, and without retaining any copies or parts thereof specification documents, data, source code or maintenance documents and all other important information and materials developed or compiled by solution provider during this project and it’s any further amendments.

3.14 Transfer of Source Code

The bidder is required to ensure that the source code generated as part of the process is transferred completely with proper file naming, comments and without tweaking. The bidder shall execute a legal agreement in specified format in this regard before the release of supply order. The bidder will need to ensure that appropriate safeguards are in place so that the information shared by SAG/DRDO is not leaked under any circumstances. No license liabilities will be entertained by SAG/DRDO on usage or alteration of the supplied software for usage in any system/subsystem of SAG/DRDO.

There shall not be any licensing issues in deploying the software (by time, place and number of copies). The same licensing terms will be applicable for any third party Software modules (Excluding open source software) if used in the customization work. Wherever open source software is used, the bidder should adhere to the licensing terms of the open source community for re-licensing and distribution & SAG will not be responsible for the same.

3.15 Review, Audit and Termination

Monthly technical reviews/audits may be conducted by buyer to determine whether the work carried out is satisfactory, meeting the requirements and ensuring timely completion of milestones. In case, progress is not found satisfactory in terms of timeliness, quality and completeness and if it fails to meet the requirements and terms and conditions detailed in this RFP, the supply order is liable for termination.

3.16 Deliverables

a) Software

i. The Software tools/applications incorporating SAG requirements as an executable as well as associated libraries, APIs & IDE.

ii. Commented source code along with complete flow chart for all software routines and subroutines.

iii. Codes and scripts for compiling, building and installing software tools along with dependencies.

iv. Codes/Software Tools/Test Data/Documents should be supplied in CD/DVD on non-returnable basis.

b) Hardware

i. Hardware for PAOIP will be delivered by the bidder with minimum specifications as mentioned in RFP.

ii. Two sample Android test device with minimum OS configuration of Android 11.0 or higher will be delivered (preferable from Google Pixel and Samsung Series) by the bidder.

c) Testing

iii. Data set used in testing various modules of the software tools as per test plan.

iv. Demonstration of the integrated PAOIP on two test devices of different manufacturers as per specification mentioned in the RFP.

d) Documents

All documents and artefacts pertaining to the s/w lifecycle should be maintained preferably in DSSD standards and delivered in both soft copies (in editable form) and hard copy (printed form). This includes following documents for the software tools:

i. Software Requirement Specification
ii. Interface Requirement Document
iii. Requirement Traceability Matrix
iv. High Level Design Document with System Architecture
v. Test Case Plan
vi. Test Report
vii. User Manual
viii. Installation Manual

3.17 Ownership of Intellectual Property (IP)

The solution provider hereby confirms and agrees that all intellectual property rights including copyright or any other rights arising from the services, or the product of the Services of the Solution Provider shall vest with DRDO as the first owner of the same pursuant to this contract of service executed.

All intellectual properties (i) conceived (whether or not actually conceived during regular business hours) or made by Solution Provider during the course of project engagement with DRDO, and (ii) ideas, techniques or principles related to the business of DRDO, shall be disclosed in writing promptly to DRDO and shall be the sole and exclusive property of DRDO and the Solution Provider hereby irrevocably assigns to DRDO, without any further consideration, his right, title and interest (throughout the world), free and clear of all liens and encumbrances, in the said Intellectual Property.

Important Note(s):

1. The software developed must be supplied with complete source code with proper comments and all supporting libraries/DLLs. There must not be any tweaking/malice at the time of delivery which makes it difficult for buyer to understand the code or hamper its functionality.
2. Demonstration for POC and detailed presentations may be sought from the bidders by the committee during evaluation.

Section 4: GANTT CHART

S.No	Months Activity	1	2	3	4	5	6	7	8	9	10	11	12
1	Requirement capture and Planning • Scope of Work • Requirement Specification Gathering • Software requirements (SRS) • Interface Requirement (IRD)	■											
2	Design • Solution/Software Architecture (SAD) of all modules • Technical Design • Design Documents	■	■ ▲	M1									
3	**Implementation** • Android Platform Integrity (AI/ML & Inference Model, Optimized Mechanism for Platform Integrity)			■	■	■ M2 ▲							
	• Patch Security Analysis(Static and Dynamic Analysis, AI/ML & Decision Models)			■	■	■	■	■ M3 ▲					
	• Integrity Impact Analysis.(Static Impact Analysis & Dynamic impact Analysis)					■	■	■	■	■ M4 ▲			
	• Central Manager(User Interface & Configuration Manager, Inference Model, Reporting & Logging)								■	■	■ M5 ▲		
4	Integration • Module Integration • Integration & Installation Manual							■	■	■	■		
5	Testing & Deployment • Unit Performance Testing • E2E Test Environment Setup/Test Plan, Acceptance Testing & Report Generation					■		■		■	■	■ M6	
6	Delivery, Documentation and Training • Delivery and Deployment • Training & Code Walk through • User manual											■	■ M7 ▲

▲ **Milestones (M1 to M7)**

Section 5: Technical Compliance Report

S.No.	Specification	Compliance (Yes/No/ Better/Info)
FUNCTIONAL REQUIREMENTS		
1.	Solution provider should develop solution for Android platform integrity such that effect of new test patch on established integrity mechanism is minimized.	
2.	Solution should provide options for the user to select a version of Android source code and targeted device configuration.	
3.	Solution provider should provide development environment to build the AOSP code which needs to be downloaded from the official website of Google.	
4.	Solution should have the sufficient database of latest patches from Android 9 onwards from publicly available databases like CVE, official Google website etc. These patches specifically needs to be security patches with high severity level patches given the most priority. Also, provision for updating patch database through GUI to be provided.	
5.	Solution should have trained AI/ML model based on the past security patches or synthetic test data generator. The effectiveness of extracted features from patch database needs to be ensured for achieving performance benchmarking (F1 score, Accuracy, Precision).	
6.	Inference model should partition the Android platform into two sets: Static Partition and Variable Partition. Solution should also ensure to separate the binaries into separate partitions along the lines of Android OS stack such as Boot partition, Kernel partition, Framework/System/HAL partition and application partition.	
7.	Solution provider should develop and implement mechanism for Android platform integrity based Hash/Hash chain, Public Key Cryptography for secure booting of smart device platform.	

	Solution should have cryptographic Safe Hashes for development of Hash based integrity Mechanism.	
8.	Testing and validation of integrity mechanism should be performed on two Smart Devices (preferable Samsung and Google Nexus Series devices) which needs to supplied by the solution provider.	
9.	Solution should store the golden hashes of the different partitions of the Android OS in a secure location with restricted user access rights.	
10.	Solution should able to perform static analysis of given test patch for the known vulnerabilities/malwares for the purpose of security analysis.	
11.	Solution should able to perform dynamic analysis of given test patch after either by applying to the test device or by running it in the emulator.	
12.	Dynamic analysis should able to detect and report difference in behaviour/functioning between Patched Smart Device and Unpatched Smart Device. Any unexpected change needs to be logged and highlighted with summary.	
13.	Solution should be able to process the inputs from static and dynamic analysis modules to deduce inference and decision about acceptability of test patch. It should be able to detect/Infer the malicious activities/abnormal behaviours and untoward incidence if any for the given test patch.	
14.	Solution should have provision to gauge the impact (static impact analysis and dynamic impact analysis) of patch on the established Android platform integrity.	
15.	Soulution should able to perform impact analysis at two levels namely one is the file level/partition level (kernel, system etc.) changes in the Android platform and other is the analysis of its effect on different applications/libraries/system calls/ inter process communication.	

16.	Solution should be able to process the inputs from static and dynamic impact analysis modules to deduce inference and decision about whether patch leads to minimal/moderate/huge changes in integrity and functionality of the smart devices along with level of associated security risk.	
17.	Solution should have User Interface to allow user to choose the Security Patch, Android OS version on which the patch needs to be applied along with the integrity mechanism for enabling the secure boot of the Android platform.	
18.	Solution should be able to perform AI/ML/Rule based analysis and decision for the results obtained through Android platform integrity and patch security analysis module.	
19.	Solution should be able to generate cumulative score about overall impact of patch on the integrity of the smart device considering parameters as security level of the patch, effect on critical applications/libraries, overall impact of patch on the integrity of the smart device.	
20.	Solution should be able to logged all the errors, results, exceptions etc. using Logging Module. It should support logging of the data in three formats - error log, warning logs & information messages.	
21.	UI Module/ Control & Configuration Manager should provide a provision to configure parameter for different test options, input data types, log and save option, report format defined/selected by user.	
22.	Solution should be able to take user configuration data either as *.xml, *.csv, *.json file or input from GUI and reconfigure.	
23.	Hardware should be from standard OEM provider with warranty and logo emboss to the product/hardware.	
24.	GUI of the solution should be simple, smooth and interactive with help guide for easy handling of the solution by the user.	
25.	Solution should work on rooted devices of the manufacturer who use stock Android, or share the code base of their	

	customized Android OS and drivers in the public domain (Open Source Kernel Code) with root access.	
26.	Solution should achieve high precision & F1 Score value for AI/ML module which is optimally trained with sufficient data.	
27.	Solution should achieve minimum accuracy of 90% for different defined inferential parameters for implemented AI/ML module.	
28.	Solution should have provision to train the AI/ML module with new user defined class of malware by invoking input data either from patch database or from test data generator.	
29.	Solution should have provisioning for generating the combined analysis results in *.pdf and Microsoft Word format.	
30.	Solution should have provision for operator to enable individual logs or all logs for the purpose of diagnosis & Information purpose.	
31.	Solution should provide features to manage users (add new users to the system, edit user details, delete users, change their permissions).	
32.	Solution should support and implemented on two test devices (preferably Samsung S Series and Google Pixel).Test device to be supplied by the solution provider.	
33.	Solution should handle any unexpected errors or exceptions within the concerned module, without causing any system instability. Errors and exceptions should be logged for further review.	
34.	Solution should execute each module independently and each module should provide output only via clearly defined interfaces. Each module should implement input sanitisation methods such that only expected values are received from other modules.	
35.	Solution should be configured to be run in a containerised environment. The containers should be configured to handle concurrent processes and should be utilising the hardware resources optimally.	

36.	Solution should integrate the software components seamlessly.	
37.	Solution should utilise stable and maintainable technologies that are widely used for the customisable code base.	
38.	Solution should be developed using only buyer approved Free and Open Source technologies, libraries and software frameworks.	
39.	Solution should be a Bundled solution consisting of both hardware and software; solution provider should share the software component details in their application making their understanding of the platform clear.	
40.	Solution should have minimum hardware configuration of 64 GB DDR4 RAM & Intel Xeon Processor with form factor of 2U Rack.	
41.	Solution should have at least 1 TB SSD capacities.	
42.	Solution should have a 2 TB storage HDD drive or higher**.**	
43.	Solution should have minimum 6 GB NVidia Graphic card.	
44.	Solution should have FHD Monitor of minimum 27″.	
45.	Solution should have a Dual power supply with a minimum 1100w fully redundant.	
46.	Solution should have USB, HDMI, and VGA ports.	
47.	Solution should be BIOS protected.	
48.	All the specified hardware should be from standard OEM.	
NON-FUNCTIONAL REQUIREMENTS		
49.	Solution may be developed on Python/Golang/Java/ Kotlin/C/C++ with UI may be developed on React JS or on angular JS.	
50.	Software coding guidelines is preferably as per DSSD standards and should be explained in the proposal with example.	

51.	Software coding guidelines include commenting style, indentation, maximum line length to be used, line breaks, blank line, import, naming conventions and other standard practices being applied to the programming languages.	
52.	The total duration of activity including training as well as ATP is 12 months from the date of placement of supply order.	
53.	The work is executed by solution provider at his own premises with all resources like manpower, hardware platforms and requisite IDEs being the sole responsibility of the solution provider.	
54.	The modular decomposition is in line with decomposition of functional requirements of PAOIP as indicated in the RFP.	
55.	Platform is bundled solution of hardware and software; solution provider is advised to share the software components detail in their bids.	
56.	Graphical User Interface includes all the functionalities of the PAOIP as specified in the RFP.	
57.	Design Consideration for performance optimization is as per the RFP.	
58.	All documents and artefacts pertaining to the s/w lifecycle should be maintained in preferably DSSD (DRDO Standard for Software Document) standards and delivered.	
59.	At least one week long training for 5 DRDO personnel will be provided at SAG Delhi/company premises.	
60.	All Indian private and Indian public sector companies are eligible to take up this project.	
61.	Companies should be in the domain of software development and testing.	
62.	Bidder provides query resolution (during the proposal evaluation) within 24 hours and that too in person visiting the SAG , DRDO office to clarify the queries or present supported documents. Failure to this may lead to disqualification of bidder.	

63.	Bidder will be required to give a technical presentation along with proof of concept and write-up on design consideration (with rationale) vis-a-vis the functional and non-functional requirements to the technical expert committee.	
64.	Acceptance criteria will be broadly based on PAOIP performance (accuracy, precision, F1 score etc.) in detection of malicious activities/abnormal behaviours and untoward incidence if any for the given test patch.	
65.	Acceptance criteria also covers acceptance for configuration, logging, user interface and dash board features.	
66.	Selected Vendor will comply the Non-disclosure agreement to protect the contents of the data and the output results from various tests before release of supply order.	
67.	Solution provider will provide warranty and maintenance of the PAOIP for one year from the project end date.	
68.	Solution provider will cover the hardware platform warranty for 3 years in their proposal.	
69.	Solution provider will return to DRDO all classified Information, in their entirely, and without retaining any copies or parts thereof specification documents, data, source code or maintenance documents and all other important information and materials developed or compiled by solution provider during this project.	
70.	Solution provider is required to ensure that the source code generated as part of the process is transferred completely with proper file naming, comments and without tweaking.	
71.	Solution provider shall ensure that there are not any licensing issues in deploying the software (by time, place and number of copies) and any third party Software modules if used in the customization work.	

72.	In case, progress is not found satisfactory in terms of timelessness, quality and completeness and if it fails to meet the requirements and terms and conditions detailed in this RFP, the supply order is liable for termination without any financial implication.	
73.	Solution provider will comply with software, hardware and documents deliverables as specified in the RFP.	
74.	Solution provider hereby confirms and agrees that all intellectual property rights including copyright or any other rights arising from the Services, or the product of the Services of the Solution provider shall vest with DRDO as the first owner of the same pursuant to this contract of service executed.	
75.	All intellectual properties (i) conceived (whether or not actually conceived during regular business hours) or made by Solution provider during the course of project engagement with DRDO, and (ii) ideas, techniques or principles related to the business of DRDO, shall be disclosed in writing promptly to DRDO and shall be the sole and exclusive property of DRDO and the Solution provider hereby irrevocably assigns to DRDO, without any further consideration, his right, title and interest (throughout the world), free and clear of all liens and encumbrances, in the said Intellectual Property.	
76.	Proposal should consist of milestone and delivery schedule with properly defined milestones.	
77.	Software requirement Document, test plan, software design document, code release and test execution time line should be defined in the proposal.	
78.	Proposal must include different project management aspects namely schedule & risk plan.	
79.	Proposal should include about approach for project management and related task.	

80.	Traceability matrix for feature mapping to requirement, design, and implementation and testing should be defined.	
81.	Proof of concept demonstration – Solution provider has to give a demo of a basic level working POC with a sample use case within the given time frame. This is important evaluation criteria for project award.	
82.	Solution provider developing this project will handover complete software and hardware including source code, requirement, design and test document to DRDO. They will destroy their copies of the available data once the project is complete; however If approved by DRDO authority, they can be allowed to keep some relevant information/code for maintenance support.	
83.	During the maintenance, solution provider may be asked to support for installation and smaller modification. However, this maintenance support does not include any major change or feature enhancement implementation.	
	Project Duration , Major Tasks/Milestones and payment schedule	
84.	Total duration of this project is 12 months from the date when project is awarded.	
85.	Development of all project requirements has to be completed within 12 months with stable operational software on specified hardware platform.	
	Project Task and milestones – Project Scope and Deliverables have been divided into 6 Tasks and 7 Milestones. T0 is start date from place of order and T0+12 is completion date (12 months from T0).	
86.	Milestone 1 (M1) at (T0+2) with Design Document – Design of Solution/ Software Architecture (SAD) of all modules, Technical Design & Design Documents.	

87.	Milestone 2 (M2) & Deliverables at (T0+5) – Android Platform Integrity with source code and realated documents (Trained AI/ML model with Database & Inference Model, Optimized Mechanisms for Platform Integrity) with Milestone 1.	
88.	Milestone 3 (M3) & Demonstration at (T0+7) – Patch Security Analysis with source code (Static and Dynamic Analysis, AI/ML & Decision Models).	
89.	Milestone 4 (M4) & Deliverables at (T0+ 9) –, Integrity Impact Analysis with source code (Static Impact Analysis & Dynamic impact Analysis) with Milestone 3 source code	
90.	Milestone 5 (M5) & Deliverable at (T0+10) – Central Manager (User Interface & Configuration Manager, Inference Model, Reporting & Logging).	
91.	Milestone 6 (M6) at (T0+11)–Integrated Solution Testing, Deployment and Acceptance.	
92.	Milestone 7 (M7) & Deliverable (T0+12) – Final Delivery of PAOIP solution (Hardware and Software along with Source Code), Documentation, User Manuals, Training & Code Walk through.	
	Payment Schedule – Below is the defined payment terms for the project execution. Bidder has to agree to this payment term and need to incorporate the same in their proposal.	
93.	15% payment will be done after complete delivery of Milestone up to 1 & 2 (deliverables up to T0+5).	
94.	15% Payment will be done after complete delivery of Milestone up to 3 & 4 (deliverables up to T0+9).	
95.	25% Payment will be done after complete delivery of up to Milestone 5 (deliverables up to T0+10).	
96.	45% Payment will be done after completion of Milestone 7 and complete solution handover along with deliverables up to T0+12.	

	Penalty Clause	
97.	Delay up to 30 days @ 0.3 % per day of the value of the milestone based payment. Beyond 45 days: may lead to cancellation of the contract subject to justification by service provider	

Section 6: Project Resource

The following persons with below mentioned experience is expected to work:

Sr. No.	Designation	Minimum Experience
1.	Lead architect (Mandatory)	min 5 years
2.	Lead Mobile Security Expert	min 8 years
3.	Lead AI/ML Expert	min 6 years
4.	Patch Analysis Expert	min 6 years
5.	Malware and Vulnerability QA (Lead)	min 6 years
6.	AI/ML Developer	min 3 years
7.	Backend/API Developer	min 3 years
8.	Frontend Developer (Desktop Client)	min 3 years
9.	UI/UX Designer	min 3 years

This is a semi-detached application by nature and required effort E ~ 71 Man Months.

Cost Estimation

S.No.	Activity	Man Months
1.	Requirement capture and Planning • Scope of Work • Requirement Specification gathering • Software requirement(SRS) • Interface Requirement(IRD)	5
2.	Design • Solution/Software Architecture (SAD) of all modules • Technical Design • Design Documents	8
3.	**Implementation** • Android Platform Integrity (AI/ML & Inference Model, Optimized Mechanism for Platform Integrity) • Patch Security Analysis(Static and Dynamic Analysis, AI/ML & Decision Models) • Integrity Impact Analysis.(Static Impact Analysis & Dynamic impact Analysis) • Central Manager (User Interface & Configuration Manager, Inference Model, Reporting & Logging)	34
4.	Integration • Module Integration • Integration & Installation Manual	12
5.	Testing & Deployment • Unit Performance Testing • E2E Test Environment Setup/Test Plan, Acceptance Testing & Report Generation	7
6.	Delivery, Documentation and Training • Delivery and Deployment • Training & Code Walk through User manual	5
	Total Man Month Cost (with GST 18%)	71*1.2=Rs 85.2 Lakhs

Average Cost per Person is @1.2 Lakhs/month

Note –This total man month cost include the platform hardware cost also.