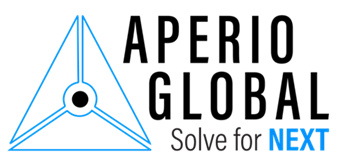
The Intelligence Advance Research Projects Activity (IARPA)

Securing our Underlying Resources in Cyber Environments  
(SoURCE CODE)

IARPA-BAA-24-02

Due: 23 May 2024

POC authorized to offer, negotiate, accept award, administer, and sign this proposed contract:

Earl Stafford, Jr.

11710 Plaza America Drive, Suite 2000

Reston, VA 20190

Phone: (703) 462-4296

Email: [estaffjr@aperioglobal.com](mailto:estaffjr@aperioglobal.com)

DUNS #: 116993065

FEIN #: 83-3643563

Cage Code #: 89JFO

Facility Clearance: TOP SECRET

Table of Contents

[1.0 Experience & Qualifications 1](#_Toc148537794)

[2.0 Technical Challenges & Objectives 1](#_Toc148537795)

[2.1 Focus Area 1 – Feature Space Generation& Extraction 1](#_Toc148537796)

[2.2 Focus Area 2 – Similarity & Demographic Analytic Algorithms 1](#_Toc148537797)

[2.3 Focus Area 3 – System Explainability 2](#_Toc148537798)

[3.0 Program Phases 2](#_Toc148537799)

[3.1 Phase 1 2](#_Toc148537800)

[3.2 Phase 2 3](#_Toc148537801)

[4.0 Program Scope & Limitations 3](#_Toc148537802)

[4.1 Underlying Theory 3](#_Toc148537803)

[4.2 Research & Development Approach 3](#_Toc148537804)

[4.3 Technical Risks 3](#_Toc148537805)

[4.4 Software Development 3](#_Toc148537806)

[5.0 Program Data 3](#_Toc148537807)

[5.1 Development Data 4](#_Toc148537808)

[5.2 Government Evaluation Data 4](#_Toc148537809)

[6.0 Personnel 4](#_Toc148537810)

[7.0 Meeting & Travel Requirements 5](#_Toc148537811)

[7.1 Technical Exchange Meetings & Workshops 5](#_Toc148537812)

[7.2 Site Visits 5](#_Toc148537813)

# Section 1: Cover Sheet & Transmittal Letter

|  |  |
| --- | --- |
| (1) BAA Number | IARPA-BAA-24-02 |
| (2) Technical Area | Securing our Underlying Resources in Cyber Environments (SoURCE CODE) |
| (3) Lead Organization Submitting Proposal | Aperio Global |
| (4) Type of Business, Selected Among the Following Categories: “Large Business”, “Small Disadvantaged Business”, “Other Small Business”, “HBCU”, “MI”, “Other Educational”, or “Other Nonprofit” | Small Disadvantaged Business |
| (5) Contractor’s Reference Number (if any) | NA |
| (6) Other Team Members (if applicable) and Type of Business for Each | George Mason University, Other Educational |
| (7) Proposal Title | MLOPS based Malware Analysis, Similarity, Attribution, |
| (8) Technical Point of Contact to Include: Title, First Name, Last Name, Street Address, City, State, Zip Code, Telephone, Fax (if available), Electronic Mail (if available) | **Damian Watkins, D.Eng**  **Chief of Research & Innovation**  **15040 Conference Center Drive Suite 150 Chantilly, VA 20151** |
| (9) Administrative Point of Contact to Include: Title, First Name, Last Name, Street Address, City, State, Zip Code, Telephone, Fax (if available), Electronic Mail (if available) | **Earl Stafford, Jr**  **CEO** 15040 Conference Center Drive Suite 150 Chantilly, VA 20151 |
| (10) Date Proposal as Submitted. | 23 May 2024 |

# Summary of the Proposal

**A. Technical Overview of the Proposed Research and Plan**

We propose leveraging the RUSSEL malware analysis system and the Cantordust tool for advanced malware detection and attribution. Our goal is to develop a robust system that enhances cybersecurity by accurately detecting and attributing malware to specific sources, improving upon current methods by integrating advanced machine learning and AI technologies.

**B. Summary of the Products, Transferable Technology and Deliverables Associated with the Proposed Research Results**

The key deliverables include:

* An integrated malware detection and attribution system combining our patented IP RUSSEL and Cantordust.
* Transferable technologies for feature extraction, similarity analysis, and explainability.
* Pre-trained models and tools for malware analysis.
* A comprehensive report on the research findings and technological advancements.

**C. Schedule and Milestones for the Proposed Research**

The project is divided into two phases:

* **Phase 1 (18 months)**: Development of feature extraction and similarity analysis techniques.
* **Phase 2 (12 months)**: Extension and integration of systems for cross-representation querying and full-scale deployment.

# Technical Challenges & Objectives

Our team brings a diverse array of expertise crucial for the SoURCE CODE program, including binary analysis, compiler design, software engineering, programming languages, authorship attribution, threat intelligence analytics, cybersecurity, cyber-forensics & reverse engineering, stylometry, artificial intelligence, and machine learning. We propose a multidisciplinary approach, integrating automated methods with expert insights from forensic practitioners.

## Focus Area 1 – Feature Space Generation& Extraction

We aim to develop advanced feature extraction techniques from source code and binary files, moving beyond lexical and syntactic features to include semantic and behavioral attributes. This will enable more robust and insightful forensic analysis. We propose leveraging the concepts from the "Learning Transferable Visual Models From Natural Language Supervision" paper, specifically focusing on the creation of robust feature spaces using large-scale pre-training on diverse datasets. By employing contrastive language-image pre-training (CLIP), we aim to generate feature representations that are both efficient and scalable. This approach involves training models to predict the pairing of images and their corresponding text descriptions from a dataset of 400 million (image, text) pairs collected from the internet​​.

## Focus Area 2 – Similarity & Demographic Analytic Algorithms

Our goal is to create algorithms that not only measure similarity between code samples but also infer demographic traits. These algorithms will utilize machine learning models trained on extensive datasets to identify patterns indicative of specific groups, countries, or individuals. Our approach includes developing similarity and demographic analytic algorithms by harnessing the pre-trained models from CLIP. These models can transfer learned features to downstream tasks such as OCR, action recognition, geo-localization, and fine-grained object classification without additional dataset-specific training​​. We will explore both supervised and unsupervised methods to enhance the accuracy and explainability of our algorithms, facilitating forensic experts in making informed decisions based on the similarity measures and demographic insights provided by the models.

## Focus Area 3 – System Explainability

Explainability is crucial for forensic experts to trust and utilize automated tools. We will incorporate explainable AI techniques to provide clear, understandable rationales for the system's outputs, ensuring that analysts can confidently interpret the results. In line with the requirements of the SoURCE CODE program, our solution will include explainability components to assist forensic experts in understanding the basis of the system's outputs. We will employ techniques from recent advances in natural language supervision to provide detailed explanations for similarity scores and demographic predictions, ensuring transparency and aiding in the validation of the system's results​​.

# Program Phases

The initial 18-month phase will focus on developing and refining feature extraction and analytic algorithms. Key deliverables will include prototype systems and performance evaluations using government-provided and supplementary datasets.

## Phase 1

The initial 18-month phase will focus on developing and refining feature extraction and analytic algorithms. Key deliverables will include prototype systems and performance evaluations using government-provided and supplementary datasets.

## Phase 2

In the subsequent 12-month phase, we will enhance the system's capabilities, addressing more complex scenarios and larger datasets. This phase will culminate in a fully operational prototype, ready for independent evaluation.

# Program Scope & Limitations

## Underlying Theory – TA1

Our approach is grounded in the Diamond Model of Forensic Attribution, focusing on the Attacker-Capability edge. We will leverage this model to inform our feature extraction and analytic techniques.

## Research & Development Approach

Our R&D strategy involves iterative development and testing, with regular feedback loops to refine our methods. We will employ open-source and proprietary tools to build and evaluate our system.

### Malware Analysis & Automated Vulnerability Assessment

Malware analysis consists of several components, all of which are active and well-documented open-source

projects. Integrating these projects will add the ability to perform an initial static analysis

on a diverse set of binary types. Zeek, an open network security monitoring tool and framework, will be

used to acquire network traffic and extract binaries of interest. In addition, Zeek provides critical network

metadata and metrics related to the given binary and additional on-network activity data of the

binary in question. In addition, the system shall allow for the ingest of binary files without using Zeek. The Ghidra Software Reverse Engineering Framework (SRE) is a full-featured software analysis

tool, and its capabilities include disassembly, assembly, de-compilation, graphing, and

scripting. Furthermore, Ghidra is used as a collaborative tool. Allowing analysts to team once

the automated analysis is complete. Yara is a security tool developed by VirusTotal and will be

used to detect malicious binaries based on textual or binary patterns.

CLIP similarity

CLIP (*Contrastive Language-image–Image Pre-training*) builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning. The idea of zero-data learning dates back over a decade but until recently was primarily studied in computer vision to generalize to unseen object categories.[9](https://openai.com/index/clip/#citation-bottom-9),[10](https://openai.com/index/clip/#citation-bottom-10) A critical insight was to leverage natural language as a flexible prediction space to enable generalization and transfer. In 2013, Richer Socher and co-authors at Stanford[11](https://openai.com/index/clip/#citation-bottom-11) developed a proof of concept by training a model on CIFAR-10 to make predictions in a word vector embedding space and showed this model could predict two unseen classes. The same year DeVISE scaled this approach and demonstrated that it was possible to fine-tune an ImageNet model so that it could generalize to correctly predicting objects outside the original 1000 training set.

Most inspirational for CLIP is the work of Ang Li and his co-authors at FAIR[13](https://openai.com/index/clip/#citation-bottom-13) who in 2016 demonstrated using natural language supervision to enable zero-shot transfer to several existing computer vision classification datasets, such as the canonical ImageNet dataset. They achieved this by fine-tuning an ImageNet CNN to predict a much wider set of visual concepts (visual n-grams) from the text of titles, descriptions, and tags of 30 million Flickr photos and were able to reach 11.5% accuracy on ImageNet zero-shot.

Finally, CLIP is part of a group of papers revisiting learning visual representations from natural language supervision in the past year. This line of work uses more modern architectures like the Transformer[32](https://openai.com/index/clip/#citation-bottom-32) and includes VirTex,[33](https://openai.com/index/clip/#citation-bottom-33) which explored autoregressive language modeling, ICMLM,[34](https://openai.com/index/clip/#citation-bottom-34) which investigated masked language modeling, and ConVIRT,[35](https://openai.com/index/clip/#citation-bottom-35) which studied the same contrastive objective we use for CLIP but in the field of medical imaging.

## Technical Risks

Key risks include the variability of coding styles and the potential for adversaries to obfuscate their code. We will mitigate these risks through robust testing and the incorporation of adaptive learning techniques.

## Software Development

Our software development process will adhere to best practices in security and scalability, ensuring that the final product is both effective and reliable.

## 4.5 MLOps Integration

To ensure the smooth and efficient deployment, monitoring, and maintenance of our machine learning models, we will integrate robust MLOps practices into our software development lifecycle. Our MLOps strategy will include:

* **Automated Pipelines:** We will develop automated pipelines for data ingestion, preprocessing, model training, evaluation, and deployment. These pipelines will use tools like Apache Airflow, Kubeflow, and Jenkins to ensure reproducibility and scalability.
* **Continuous Integration/Continuous Deployment (CI/CD):** Our CI/CD framework will enable seamless updates and deployments of models and software components. We will use containerization technologies such as Docker and orchestration tools like Kubernetes to manage and deploy our services efficiently.
* **Monitoring & Logging:** We will implement comprehensive monitoring and logging solutions using tools like Prometheus, Grafana, and ELK Stack (Elasticsearch, Logstash, and Kibana). This will allow us to track the performance, reliability, and health of our models and systems in real-time, enabling quick identification and resolution of issues.
* **Versioning & Reproducibility:** Our MLOps framework will ensure version control for data, models, and code using tools like DVC (Data Version Control) and Git. This will facilitate reproducibility and traceability, allowing us to roll back to previous versions if needed and maintain a clear audit trail of changes.
* **Model Governance:** We will establish governance policies for model lifecycle management, including regular audits, compliance checks,

## Underlying Theory – TA2

## Underlying Theory – TA3