HBGary

HBGary Federal, LLC.
3604 Fair Oaks Blvd. Suite 250, Sacramento, CA. 95864
Phone: (916) 459-4727   Fax: (916) 481-1460

VOLUME I TECHNICAL MANAGEMENT PROPOSAL

Prepared for DARPA

CYBER GENOME PROGRAM

STRATEGIC TECHNOLOGY OFFICE

DARPA-BAA-10-36

March 21, 2010

This proposal includes data that shall not be disclosed outside the Government and shall not be duplicated, used, or disclosed-in whole or in part-for any purpose other than to evaluate this proposal. If, however, a contract is awarded to this proposer as a result of, or in connection with, the submission of this data, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the resulting contract. This restriction does not limit the Government's right to use information contained in this data if it is obtained from another source without restriction. The data subject to this restriction are contained in marked sheets.

Table of Contents

Section I. Administrative

A. Proposal Cover Sheet

1	**Broad Agency Announcement**	*DARPA-BAA-10-36* *Cyber Genome Program*	
2	**Prime Organization**	*HBGary Federal, LLC.*	
3	**Proposal Title**	*Cyber Genome Program, Cyber Physiology*	
4	**Type of Business (Check one)**	□ Large Business □ Small Disadvantaged Business X Other Small Business □ Government Laboratory or FFRDC	□ Historically-Black Colleges □ Minority Institution (MI) □ Other Educational □ Other Nonprofit
5	**Contractor's Reference Number**		
6	**Contractor and Government Entity (CAGE) Code**	*5U1U6*	
7	**Dun and Bradstreet (DUN) Number**	*832950831*	
8	**North American Industrial Classification System (NAICS) Number**	*541512*	
9	**Taxpayer Identification Number (TIN)**	*27-1485507*	
10	**Technical Point of Contact**	*Mr. Aaron Barr, CEO, HBGary Federal* *3604 Fair Oaks Blvd B STE 250* *Sacramento, CA 95864*	
11	**Administrative Point of Contact**	*Mr. Ted Vera, President, HBGary Federal* *3604 Fair Oaks Blvd B STE 250* *Sacramento, CA 95864*	
12	**Security Point of Contact**	*Mr. Aaron Barr, CEO, HBGary Federal* *3604 Fair Oaks Blvd B STE 250* *Sacramento, CA 95864*	

13	**Other Team Members (if applicable)**	*Pikewerks, Other Small SecureDecisions, Other Small*	*Technical POC salutation, last name, first name, street address, city, state, zip code, telephone, fax (if available), electronic mail (if available), CAGE Code*

14	**Funds Requested From DARPA**	**Base Effort: (Phase 1)**	*Base Effort Cost*
			Base Options Cost: (list all)
		Option Effort: (Phase 2)	*Option Effort Cost*
			Phase II Options Cost: (list all)
		Total Proposed Cost (Including Options)	*Total*
		Amount of Cost Share	*Amount of cost share (if any)*
15	**Award Instrument Requested**	Xcost-plus-fixed-fee □cost-contract-no-fee □cost sharing contract-no fee □other procurement contract:______________	□grant □agreement □other award instrument: __________
16	**Proposers Cognizant Government Administration Office**	*Name, mailing address, telephone number and Point of Contact of the Proposers cognizant government administration office (i.e., Defense Contract Management Agency (DCMA))*	
17	**Proposer's Cognizant Defense Contract Audit Agency (DCAA) audit Office**	*Name, mailing address, telephone number, and Point of Contact if known*	
18	**Other**	*Any Forward Pricing Rate Agreement, other such Approved Rate Information, or such other documentation that may assist in expediting negotiations (if available)*	
19	**Date Proposal Prepared**	*Mar. 29, 2010*	
20	**Proposal Expiration Date**	*July 30, 2010*	
21	**Place(s) and Period(s) of Performance**	*Location where the proposed work will be performed and dates of proposed performance*	
22	**Technical Area (check one)**	□ Technical Area 1 - Cyber Genetics □ Technical Area 2 - Cyber Anthropology and Sociology X Technical Area 3 - Cyber Physiology □ Technical Area 4 – Other	

B. *Official transmittal letter.*

HBGary

HBGary Federal, LLC.
3604 Fair Oaks Blvd. Suite 250, Sacramento, CA. 95864
Phone: (916) 459-4727   Fax: (916) 481-1460

March 29, 2010

Attn: Dr. Michael VanPutte
Defense Advanced Research Projects Agency

Subject: DARPA Cyber Genome Program

HBGary Federal is pleased to present this proposal to DARPA in response to DARPA BAA-10-36 Cyber Genome Program Technical Area III: Cyber Physiology. This proposal assumes a CPFF type contract and is valid through July 30, 2010.

Cost
Fixed Fee
Total CPFF

Organizational Conflict of Interest

HBGary Federal, LLC. does not provide scientific, engineering and technical assistance (SETA) or similar support to any DARPA technical office(s) through active contracts or subcontracts. We therefore do not have any organizational conflicts of interest that require affirmation.

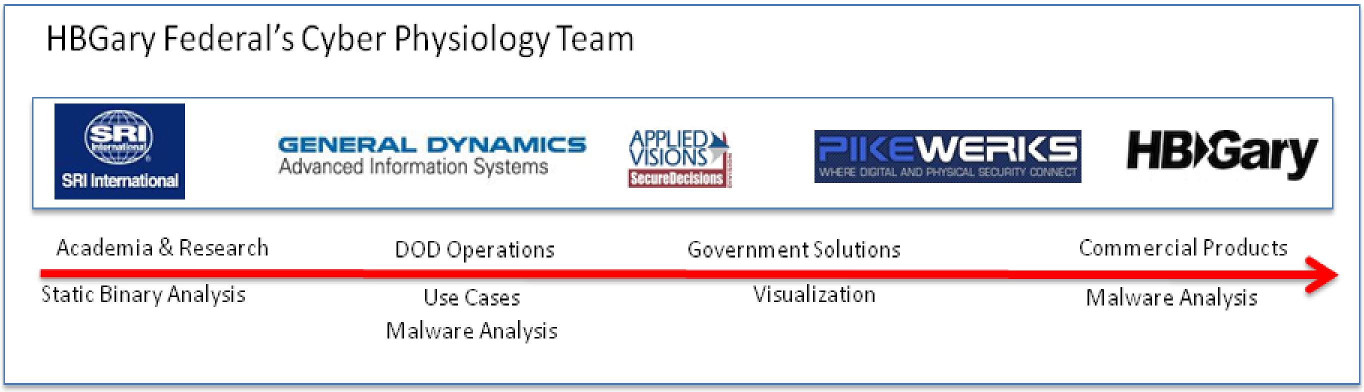
Sincerely yours,

Aaron D. Barr
CEO
HBGary Federal, LLC.

Section II. Summary of Proposal

II.A Innovative Claims for the Proposed Research

Our HBGary Federal Team comprises some of the most capable companies and research organizations in the field of malware analysis and visualization. Together, we offer a revolutionary approach to addressing Technical Area Three, Cyber Physiology that builds on our depth and breadth of experience. From research to product to operations, we all are documented leaders in our fields, with demonstrated capabilities to provide cyber defense and investigatory technologies in support of defense, law enforcement, and intelligence and counter intelligence



In our proposed Cyber Physiology system, malware objects are pre-processed to remove obfuscation and anti-analysis capabilities, then stored in the specimen repository and flagged for execution and analysis. A combination of memory and runtime analysis is performed using the developed traits and patterns libraries and data flow tracing used to collect near full execution of all code and low-level data and stored back into the repository, a physiology profile is developed that mathematically and descriptively represents the malware aggregate functions, behaviors, and intent. A Physiology Profile report can be generated through our visualization interface, which shows a variety of graphical representations of the malware object and allows an analyst to interact with the models to better understand. Once mature data sets exist there will be a capability to process the low level data outputs from the memory and runtime analysis through a reasoning engine that can make probability decisions on malware functions and behaviors even for previously undefined traits and patterns.

Table 1. Innovative Claims for the Proposed Research

Research Area	Innovative Claim	State-of-the-Art
Specimen Collection and Pre-Processing	The most advanced binary unpacking and automated de-obfuscation system. Self-evaluation metrics will allow it to iteratively detect and recover from binary unpacking problems and avoid anti-reverse engineering countermeasures It will incorporate snapshot-stitching techniques to deal with multi-stage packers and block encryption. We will research and develop automated ways to recognize obfuscated code and identify the obfuscation steps employed to hinder automated analysis, then systematically de-obfuscate to restore the binary to an equivalent but un-obfuscated form.	Current de-obfuscation techniques are not fully automated, and cannot resolve APIs automatically, nor reliably auto-discover the original entry point. They cannot deal with block encryption or code segmentation. Current binary unpacking systems are tuned toward static disassembly and analysis. These systems yield a disassembled approximation of the binary that does not support logic and data flow extraction through the informed execution of malware.

Specimen Analysis and Visualization	Visual representations of malware, through analyst views and the **Cyber Physiology Profile**, that allow for easy understanding of the malware behaviors, functions, and intent.	A few capabilities that show loop and branch and function view of malware, but they only view, without any functional context or purpose.
Traits Library	A comprehensive data set that describes the discrete functions and behaviors of malware through mathematical representations, rule sets, and descriptions.	Limited capabilities/tools that describe some subset of discrete functions and behaviors of malware but not in a standardized, comprehensive manner that can be mathematically calculated and automated.
Genomes Library	A library that codifies complex patterns within malware that indicates aggregate functions and behaviors. This is the heart of what is missing today.	Some theory and research papers exist that discuss the potential benefits of codifying complex patterns of functions and behaviors of malware
Static Malware Analysis and Runtime Tracing	An integrated and automated approach to capturing nearly 100% of code coverage of an analyzed malware object using memory and runtime analysis.	Most capabilities still exist in manual dissasemblers and interactive debuggers. No existing automated capability to combine memory and runtime data for full code path resolution.
Belief Reasoning and Inference Network	Using reasoning models, deliver a completely automated capability to analyze malware and discern behaviors and functions for previously unidentified traits and genomes.	No existing capability to define unknown characteristics of malware. Research that describes the potential benefits of using machine learning and reasoning engines for malware analysis.

II.B Deliverables, Plans, and Capability for technology transition and Commercialization

II.B.1 Deliverables

In the course of this Cyber Genome Project the HBGary Federal team will make regularly scheduled deliveries to the Government including but not limited to the following:

- Monthly reports detailing current research to include
  - Written use cases and investigation plans
  - Software architectural diagrams and algorithms
  - Source code and executable machine code of prototypes developed
- On a less frequent basis and at DARPA's direction the team will deliver detailed presentations of work progress and conduct software prototype demonstrations.
- Research Papers for each of the research areas
- Data and Libraries for Traits and Genomes
- Prototypes for malware object pre-processor, visualization interface, memory and runtime tracing, and reasoning engine

II.B.2 Plans and Capability to Achieve Commercialization and Technology Transition

HBGary and Pikewerks have track records of commercialization success. They have successfully transitioned their cyber security software products to the operational environment, as evidenced by hundreds of active customers. These were developed in part via the Small Business Innovative Research program. If awarded the contract, we anticipate that promising technologies will emerge from our research that will be desired by both Government and private sector organizations. Where appropriate, we will offer the technologies to the Department of Defense (DoD), the Intelligence Community (IC) and civilian agencies for further development and transition to operations. But we will not rely on the Government for technology transition. We anticipate making significant additional IRAD investment to convert the results of this contract into commercial grade software.

II.B.3 Data Rights and Intellectual Property

HBGary has developed two patented technologies that it brings to the table for possible use to fulfill this requirement -- Digital DNA Sequence and Fuzzy Hash Algorithm. We propose these technologies for *possible* use to fulfill this requirement; although it is possible these technologies may end up playing no role in developing the methodology that DARPA seeks. At the very least, the team will leverage the tremendous experience gained in developing these two technologies. If and to the extent that these two technologies become deliverables in the resulting contract, HBGary will deliver them with Limited Rights. (See table below). To the extent that any modifications to these two existing, proprietary technologies need to be made, HBGary will perform such modifications under pre-existing administrative codes billed to HBGary indirect accounts, and they will not be charged under the contract.

Table 2: Existing Intellectual Property Table

Assertion of Technical Data Rights in accordance with DFARS 252.227-7018			
Technical Data Computer Software To be Furnished With Restrictions	Basis for Assertion	Asserted Rights Category	Name of Person Asserting Restrictions
Digital DNA Sequence	Developed at Private Expense	Limited Rights	Bob Slapnik, Vice President HBGary, Inc.
Fuzzy Hash Algorithm	Developed at Private Expense	Limited Rights	Bob Slapnik, Vice President HBGary, Inc.
HBGary Digital DNA ™ commercial software (1)	Developed at Private Expense	Limited Rights	Bob Slapnik, Vice President HBGary, Inc.
HBGary Responder™ Professional commercial software (1)	Developed at Private Expense and SBIR, non-severable	Limited Rights	Bob Slapnik, Vice President HBGary, Inc.
HBGary REcon ™ commercial software (1)	Developed at Private Expense and SBIR, non-severable	Limited Rights	Bob Slapnik, Vice President HBGary, Inc.
Eureka	Developed with mixed funding	Government Purpose Rights	SRI

(1) Data involved in and related to commercial software products listed above will not be delivered nor do they need to be delivered to fulfill the requirements of this BAA contract, if awarded, but will be discussed in the proposal.

Digital DNA Sequence

The digital DNA sequencing engine is a system or method to evaluate any data object received via any device, network or physical memory based upon a set of rules ("genome"). The invention evaluates the contents of the digital object and generates a digital DNA sequence, which permits the data object to be classified into an object type. A trait has a rule, weight, trait-code, and description. A DDNA sequence is formed by at least one expressed trait with reference to a particular data object that has been evaluated by the DDNA engine. Typically, a DDNA sequence is formed by a set of expressed traits with reference to a particular data object that has been evaluated by the DDNA engine. When a rule fires, then that means that the trait code (or trait) for that rule has been expressed. In an embodiment of the invention, the traits can be concatenated together as a single digital file (or string) that the user can easily access.

- Patent application number: 12/386,970
- Inventor name(s): Michael Gregory Hoglund

- Assignee names: HBGary, Inc.
- Filing date: April 24, 2009
- Filing date of any related provisional application: not applicable
- Summary of the patent title: Digital DNA Sequence

HBGary's ownership of the invention is indicated in Reel/Frame 023009/0815 in the Assignment Division of the US Patent and Trademark Office.

Fuzzy Hash Algorithm

An embodiment of the invention provides an algorithm that will generate a fuzzy hash value to identify contents of a data object and to classify a data object. A digital DNA sequencing engine may be used to execute the fuzzy hash algorithm. A fuzzy hash value is a calculated sequence of bytes (e.g., hexadecimal bytes). A data stream is data content of a data object. The algorithm will place meta-tags (i.e., metadata tags) in a buffer, where a meta-tag corresponds to a value in the data stream. The fuzzy hash value can be calculated against varied data streams and can then be used to determine the percentage of match between those data streams.

- Patent application number: 12/459,203
- Inventor name(s): Michael Gregory Hoglund
- Assignee names: HBGary, Inc.
- Filing date: June 26, 2009
- Filing date of any related provisional application: not applicable
- Summary of the patent title: Fuzzy Hash Algorithm

HBGary's ownership of the invention is indicated in Reel/Frame 023441/0496 in the Assignment Division of the US Patent and Trademark Office.

II.C Cost, Schedule and Measurable Milestones

for the proposed research, including estimates of cost for each task in each year of the effort delineated by the prime and major subcontractors, total cost and company cost share, if applicable. **Note: Measurable milestones should capture key development points in tasks and should be clearly articulated and defined in time relative to start of effort.** These milestones should enable and support a decision for the next part of the effort. Additional interim non-critical management milestones are also highly encouraged at a regular interval.

*Recommend metrics that we strive to achieve in phase 1 and phase 2 in order to demonstrate technological progress. Cite quantitative and qualitative success criteria that the proposed technology will achieve by the time of each phases program metric measurement, as well as explain how the proposed effort will achieve those criteria.

Table 3. Program Costs by Company and Year

Company	Phase 1a	Phase 1b	Phase 2a	Phase 2b	Total
HBGary Federal	$500,000	$500,000	$500,000	$500,000	$2,000,000
HBGary	$300,000	$400,000	$400,000	$400,000	$1,500,000
Pikewerks	$516,168	$532,654	$548,070	$386,862	$1983,753
SRI	$499,997	$499,925	$0	$0	$999,922
Secure Decisions	$435,937	$465,727	$0	$0	$801,664
General Dynamics	$176,971	$188,470	$166,180	$170,920	$702,541
Total	$2,429,073.00	$2,586,776.00	$1,614,250.00	$1,457,782.00	$7,987,880.00

Table 4. Task Costs by Company and Year

Task	Contractor	Year	Cost	Success Criteria
Task1	SRI	1	$499,997	developed techniques for automating unpacking, de-obfuscating, and mitigating anti-analysis techniques achieved through research.
	Pikewerks		$326,083	Working prototypes and techniques for collecting Linux-based malware in the wild.
			$826,080	
	SRI	2	499,925	Developed prototypes that successfully unpack/de-obfuscate, and mitigate anti-analysis techniques on over 50% of malware employing such techniques
	Pikewerks		229,100	Mature and robust capabilities for collecting Linux-based malware
			$729,025	
	Pikewerks	3	$119,227	Enhanced collection methods for Linux-based malware
	Pikewerks	4	$89505	Enhanced collection methods for Linux-based malware
	Total Task 1		**$1,763,837**	
Task 2	HBGary Federal	1	$50,000	Developed database architecture with appropriate schema for storing all related malware specimen data, including; object, traits, genomes, analysis and tracing meta-data, and physiology profile
	Total Task 2		**$50,000**	
Task 3	Secure Decisions	1	$435,937	Proof-of-concept visualizations of malware behavior, function, and structure that enhance understanding and identification of malware characteristics
	GDAIS		$26,119	Provide relevant use cases that aid in the development of visualizations of malware
			$462056	
	Secure Decisions	2	$465,727	Enhanced prototype visualizations of malware overall behavior and functions as well as more detailed views of traits and patterns that enhance manual analysis and overall understanding of malware behavior, function, and intent.
	GDAIS		$26789	Provide relevant use cases that aid in the development of visualizations of malware
			492,516	
	Total Task 3		**$954,572**	
Task 4	HBGary Federal	2	$	Proof-of-concept foundational Windows-based genomes library that can be applied during malware analysis to identify trait patterns unique to malware
	HBGary			Support the successful development of malware genomes (complex trait patterns: sequences, clusters, conditional)
	Pikewerks		$52,346	Proof-of-concept foundational Linux-based genomes library that can be applied during malware analysis to identify trait patterns unique to malware
			$0	
	HBGary Federal	3		
	HBGary			
	Pikewerks		$119,227	
			$0	
	HBGary Federal	4		
	HBGary		$0	
	Pikewerks			
			$0	
	Total Task 4		**$0**	
Task 5	HBGary Federal	1	$350,000	Proof-of-concept foundational traits library that can be applied during malware analysis to identify and qualify traits that represent discrete functions and behaviors in malware
	HBGary		$250,000	Support the successful development of malware traits
	Pikewerks		$118,369	Proof-of-concept foundational Linux-based traits library that can be applied during malware analysis to identify and qualify traits that represent discrete functions and behaviors in malware

	General Dynamics		$80,366	Support the successful development of malware traits
			$0	
	HBGary Federal	2		Prototype malware traits library that successfully identifies malware discrete behaviors and functions based on trait matches.
	HBGary			Support the successful development of malware traits
	Pikewerks		$52,346	Prototype malware Linux-based traits library that successfully identifies malware discrete behaviors and functions based on trait matches.
	General Dynamics		$82,428	Support the successful development of malware traits
			$0	
	HBGary Federal	3		Mature malware traits library to decrease false positives and increase accuracy of identification of malware discrete behaviors and functions
	HBGary			Support the successful development of malware traits
	Pikewerks		$119.227	Mature malware Linux-based traits library to decrease false positives and increase accuracy of identification of malware discrete behaviors and functions
	General Dynamics		$84,795	Support the successful development of malware traits
			$0	
	HBGary Federal	4		Mature malware traits library to decrease false positives and increase accuracy of identification of malware discrete behaviors and functions
	HBGary			Support the successful development of malware traits
	Pikewerks		$122,804	Mature malware Linux-based traits library to decrease false positives and increase accuracy of identification of malware discrete behaviors and functions
	General Dynamics		$87,235	Support the successful development of malware traits
			$0	
	Total Task 5		**$0**	
Task 6	HBGary	2		
	Pikewerks		$129,224	
			$0	
	HBGary	3		
	Pikewerks		$119,227	
			$0	
	HBGary	4		
	Pikewerks		$122,804	
			$0	
			$0	
Task 7	HBGary Federal	3		
	HBGary Federal	4		
			$0	

II.D Technical Rationale, Technical Approach, and Constructive Plan

II.D.1 Technical Rationale

While it is a challenging undertaking, we plan to research and develop a fully automated malware analysis framework that will produce results comparable with the best reverse engineering experts, and complete the analysis in a fast, scalable system without human interaction. In the completed mature system, the only human involvement will be the consumption of reports and visualizations of malware profiles.

Our approach is a major shift from common binary and malware analysis today, requiring manual labor by highly skilled and well-paid engineers. Results are slow, unpredictable, expensive and don't scale. Engineers are required to be proficient with low-level assembly code and operating system internals. Results depend upon their ability to interpret and model complex program logic and ever-changing computer states. The most common tools are disassemblers for static analysis and interactive debuggers for dynamic analysis. The best engineers have an ad-hoc collection of non-standard homegrown or Internet-collected plug-ins. Complex malware protection mechanisms, such as packing, obfuscation, encryption and anti-debugging techniques, present further challenges that slow down and thwart traditional reverse engineering technique.

We start with the realization that malware is just software in binary form without source code. Like any software, malware must execute to do what it does. To execute it must reside in physical memory (RAM) and be operated on by the CPU. The CPU has two requirements: 1) the operating instructions of the binary must be in clear text, and 2) the CPU does only one thing at a time. A binary that is packed or encrypted must unpack or unencrypt itself; otherwise the CPU will not operate on it.

We will solve the problems with traditional reverse engineering by running the binary in a controlled, instrumented and automated run trace system that will harvest everything the CPU does, one operation at a time in sequential fashion. All instructions and data will be collected and stored in the exactly the same sequence as they occur. Replaying the execution will reproduce the binary's behaviors, along with contextual information about interactions with other digital objects. Physical memory can be imaged and automatically reconstructed, revealing all digital objects in memory at that point in time. The binary can be extracted from the memory image – typically unpacked and unencrypted – and analyzed statically, along with the contextual information contained within the memory image. From the automated run tracing and memory reconstruction we will have harvested and collected vast amounts of low-level data about the binary under test.

We make the assumption that there is a finite set of possible functions and behaviors that software and malware can have, although it can be a large set as software evolves over time. For example, there are only so many ways to communicate over the network, to survive reboot or to write to a file. We will create a set of traits and genomes that predefine observable functions and behaviors of software and malware. Using a set of rules to operate on the vast low level data collected from the binary run trace and memory reconstruction, the system will automatically determine which traits and genomes exist in each binary sample. Over time, this approach will also be able to determine evolutionary changes in the traits and genomes.

Even though the automated analysis has moved from granular technical data to the higher levels of traits and genomes, this level of information is insufficient to completely describe the functions, behaviors and intent of the binary sample. The observed traits and genomes will be fed into the Belief Reasoning engine that uses prior knowledge to make probabilistic decisions about the binary. The user will be presented with visual representations of malware physiology profiles.

II.D.2 Technical Approach and Constructive Plan

Fig. 1 illustrates our malware analysis framework, which will allow users to quickly comprehend malware functions, behaviors and intent in a **fully automated system**. The system will automatically recognize traits and genomes to classify and categorize binaries and malware. During the initial phase, traits and genomes will be developed manually, but ultimately the mature system will create traits and genomes automatically during later phases based on prior knowledge of malware. The mature system will rely on manual development of traits and genomes only as an exception. The low-level data generation will occur using an iterative static memory and runtime tracing approach. The three data sets – the Malware Specimen Repository, Traits and Genomes Libraries – will be continually updated with data through the analysis process, to include a resulting malware physiology profile. The physiology profile will contain mathematical and visual representations of the malware, as well as a human readable summary of the malware's overall and more detailed behaviors, functions, and purpose.

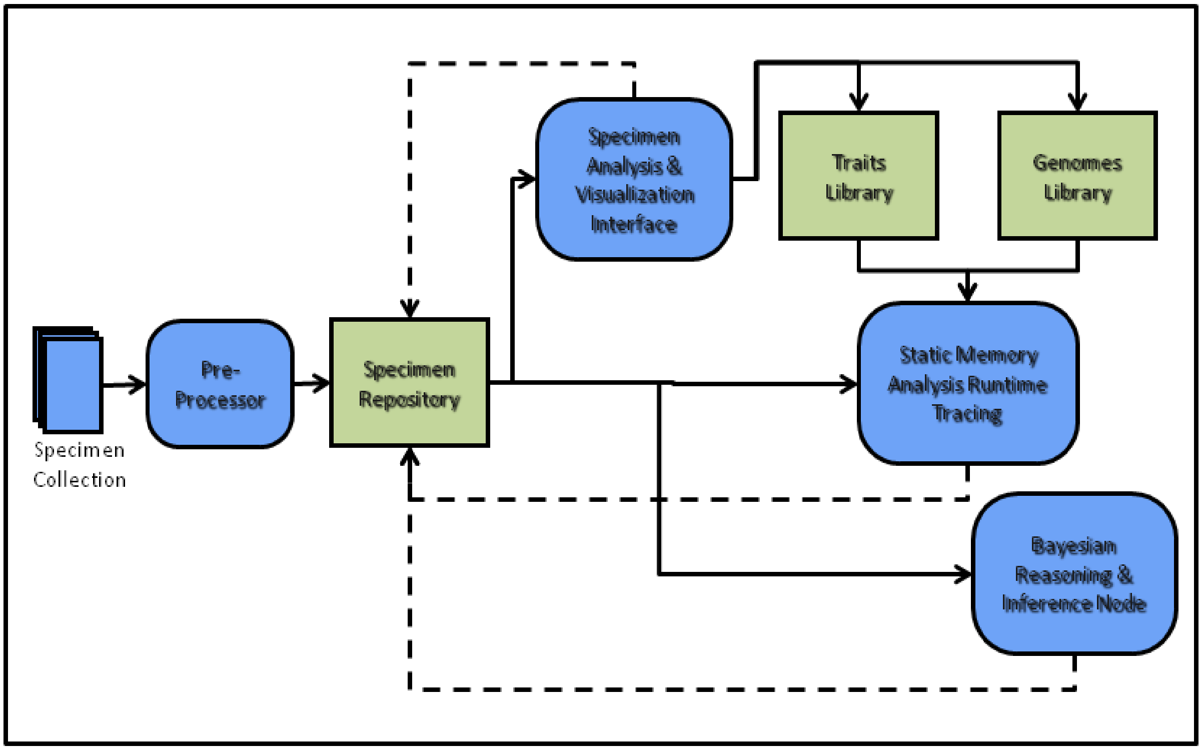

Specimen Analysis & Visualization Interface
Traits Library
Genomes Library
Pre-Processor
Specimen Repository
Static Memory Analysis Runtime Tracing
Specimen Collection
Bayesian Reasoning & Inference Node

Fig.1: Cyber Physiology Analysis Framework

Cyber Physiology Analysis Framework:

1. Specimen Collection and Pre-Processing – Subscriptions to malware feeds for updated malware objects. We also propose to research methods for identifying and collecting emergent malware specimens that are less common than the traditional Windows binary malware. For Pre-processing, we will research automated and comprehensive methods for static binary preparation, external analysis, and instrumentation, including: unpacking, de-obfuscating, reconstructing, removing anti-analysis mechanisms, and discovering environmental triggers. The goal of this phase is to normalize and prepare malware specimens for automated memory analysis and runtime tracing.
2. Specimens Repository – The central repository for specimen objects, as well as analytical information collected during pre-processing and the analysis process, with all of the memory data related to the specimen, low-level data collected during runtime tracing, and the final physiology profiles. The goal of this phase is to create a single malware repository that contains sufficient data, organized to improve malware analysis and incident response capabilities as well as integrate easily with malware lineage capabilities. HBGary brings an existing malware repository, approximately 500GB of unique malware samples to start the effort. We will conduct research for data format normalization and standardization for malware analysis results. Information maintained will include: specimen raw files, hard artifacts, associated traits and genomes, all low level data recorded through static and runtime analysis, and a full malware physiology profile.
3. Specimen Analysis & Visualization Interface (SAVI) – Methodology for streamlined analysis to assist in identifying new traits and genomes, as well as present malware physiology profiles. Research will focus on visual representations of malware data to aid in analysis and understanding of malware's functions and

behaviors and purpose. When there are function and behavior traits or genome sequences that are not fully understood by the automated system, those are flagged in the malware physiology profile stored in the specimen repository and scheduled for manual analysis.

4. Traits (Gene) Library – A repository of developed trait rules that represent discrete functions, behaviors, and intent of software. To best understand the aggregate functions, behaviors, and purpose of malware, we propose to first identify and understand the discrete expressed parts of malware at their lowest level and build up, qualifying them in a way that can be classified and mathematically calculated.
5. Genomes Library – A repository of identified trait patterns and sequences that express an aggregated functionality or behavior. These algorithms and patterns will be used to develop the visual and mathematical graphs that highlight the malware's overall function, purpose, severity. The sequences, ordering, and clustering of traits will support development of behavior and function correlation engines and visual representations based on exhibited traits, including external and environmental artifacts, space and temporal artifact relationships, and sequencing.
6. Static Memory Analysis and Runtime Tracer (SMART) – Uses a combination of static memory analysis and runtime tracing techniques to collect and record as much of the malware internals as possible, including exercising as much of the full execution tree as possible. Our research will focus on full branch execution, as well as automated analysis and tracing. HBGary and Pikewerks have existing semi-automated technologies that we can leverage for the research and development in this task.
7. Belief Reasoning Analysis and Inference Node (BRAIN) – We should be able to instrument a Belief Reasoning Engine to automatically identify mutations within the genomes and classify those mutations to some degree without any manual analysis. Our research will focus on building the malware behavior and function inference models to do the automated analysis of malware.

II.E Detailed Management, Staffing, Organization Chart, and Key Personnel:

As a small business, HBGary Federal has a very simple and streamlined approach to program management, defining a framework for the research and development with well-defined responsibilities and interfaces for collaboration, and exchange of information. This includes a detailed research and development schedule. The program quantitative and qualitative success criteria will be included in the schedule, milestones, and deliverables, with progress updated regularly in weekly management and technical discussions. The Principle Investigator is responsible for the overall technical direction of the effort and quality of the technical deliverables, and as such will lead the technical approach, make decisions on redirection based on research results measured against the quantitative and qualitative success criteria. The Program Manager is responsible for the cost and schedule of the effort and works closely with the Principle Investigator to ensure the team is meeting the technical, quantitative and qualitative goals of the effort within the cost and schedule proposed. Each of the subcontractor provides an individual responsible for leading their areas of responsibility within the project (listed below as Key Personnel).

II.E.1 Management

HBGary Federal will manage all project deliverables through all execution phases of this contract and will hold weekly Technical and Management meetings with the research leads (key personnel) or representative of each the team members to ensure we are managing cost, schedule and milestones in meeting quantitative and qualitative success criteria.

II.E.2 Teaming and Staffing

HBGary Federal's teaming strategy focuses on addressing the hard problems associated with automated analysis of malwares behavior, function, and intent. Our team offers the companies with the most significant capabilities to research, develop, and ***deliver*** tangible, quantitative and qualitative solutions. This requires organizations with extensive experience in malware research, binary instrumentation, cyber security operations and investigations, computer security productizing, malware analysis products and services, visualization, data management, and Windows and Linux malware analysis. We are very proud of our team, which we believe offer the greatest depth and breadth of experience in each of these essential areas of focus.

II.E.3 Organizational Chart

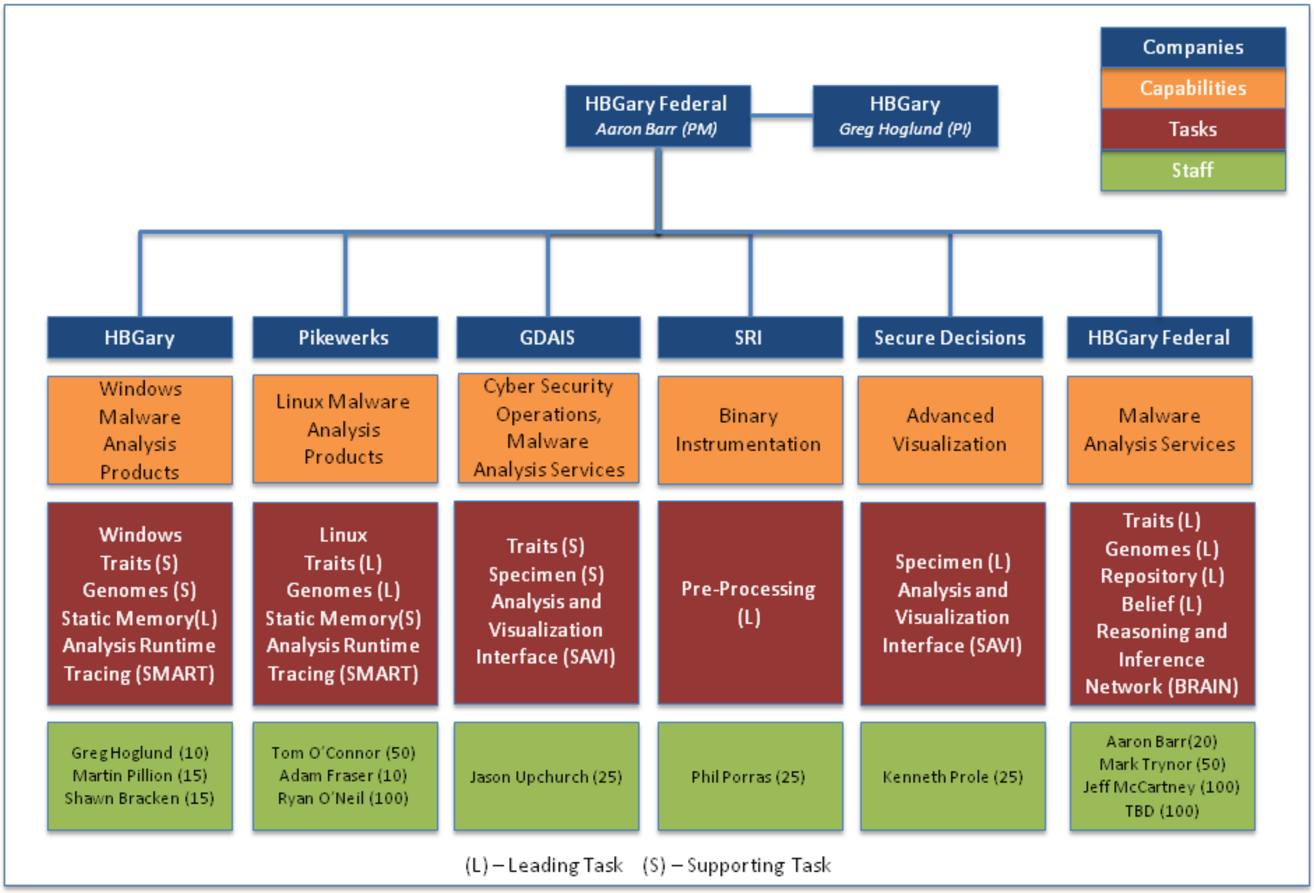

Companies
Capabilities
Tasks
Staff
HBGary Federal
Aaron Barr (PM)
HBGary
Greg Hoglund (PI)
HBGary
Pikewerks
GDAIS
SRI
Secure Decisions
HBGary Federal
Windows Malware Analysis Products
Linux Malware Analysis Products
Cyber Security Operations, Malware Analysis Services
Binary Instrumentation
Advanced Visualization
Malware Analysis Services
Windows Traits (S) Genomes (S) Static Memory(L) Analysis Runtime Tracing (SMART)
Linux Traits (L) Genomes (L) Static Memory(S) Analysis Runtime Tracing (SMART)
Traits (S) Specimen (S) Analysis and Visualization Interface (SAVI)
Pre-Processing (L)
Specimen (L) Analysis and Visualization Interface (SAVI)
Traits (L) Genomes (L) Repository (L) Belief (L) Reasoning and Inference Network (BRAIN)
Greg Hoglund (10) Martin Pillion (15) Shawn Bracken (15)
Tom O'Connor (50) Adam Fraser (10) Ryan O'Neil (100)
Jason Upchurch (25)
Phil Porras (25)
Kenneth Prole (25)
Aaron Barr(20) Mark Trynor (50) Jeff McCartney (100) TBD (100)
(L) – Leading Task (S) – Supporting Task

Fig 2. Organizational Chart

II.E.4 Key Personnel

Greg Hoglund, Chief Executive Officer	
Proposed Role:	Principal Investigator
Company	HBGary Inc.
Proposed Level of Support:	15%
Location:	Sacramento, California
Greg Hoglund is a world renowned cyber security and Windows internals expert. He architected HBGary's commercial cyber security software products Digital DNA, Responder and REcon. He pioneered new technologies to automatically reverse engineer software binaries from within computer memory and technologies to automatically harvest malware behaviors during its execution. Greg has published many significant works in the cyber security field, including: *Rootkits: Subverting the Windows Kernel*; *Exploiting Software: How to Break Code*; *Exploiting Online Games;Hacking World of Warcraft: An Exercise in Advanced Rootkit Design*; *VICE - Catch the Hookers!;Runtime Decompilation; Exploiting Parsing Vulnerabilities;Application Testing Through Fault Injection Techniques;Kernel Mode Rootkits; Advanced Buffer Overflow Techniques; A *REAL* NT Rootkit, patching the NT Kernel*.	

He created and documented the first Windows kernel rootkit, owns the rootkit forum (http://www.rootkit.com) and created a popular training program “Offensive Aspects of Rootkit Technology.” Greg has mastery in software design and development, software reverse engineering, network protocols, network programming, and packet parsing. He is fluent and highly experience with developing Windows device drivers, debuggers and disassemblers. Prior to founding HBGary, Greg was founder and CTO of Cenzic where he developed Hailstorm, a software fault injection test tool.

Aaron Barr, Chief Executive Officer

Proposed Role:	Program Manager
Company	HBGary Federal, LLC.
Proposed Level of Support:	20%
Education:	M.S. Computer Science
Location:	Washington, DC

Aaron Barr has seven years of program management experience at increasing levels of responsibility. Most recently he was responsible for developing and implementing Northrop Grumman’s Cyber and SIGINT Systems Business Unit technical strategy and ensuring quality technical execution on programs. He provided input to key targets and technical approaches to the LRSP and AOP of a $700M organization. His responsibilities included managing a $20M R&D program across Cyber, SIGINT, Airborne, and Special Access Programs.

Aaron was also the Chief Engineer for Northrop Grumman Corporations cyber security Integration Group, developing the technical cyber security strategy for the company.

Tom O’Connor, Principle Investigator

Proposed Role:	Research Lead
Company	Pikewerks
Proposed Level of Support:	100%
Education:	B.S. Physics & Computer Science
Location:	Washington, DC

Tom O’Connor has over ten years experience in software development on multiple platforms. Tom has contributed to the development of software security products in both the government funded research and commercial sectors. After graduating from William & Mary in 1997, he joined the research team at Cigital (formerly Reliable Software Technologies). At Cigital, he focused on developing source-based software security tools for both C and Java. Results of Tom's research into using fault injection to identify software security flaws were presented at the 1998 IEEE Symposium on Security & Privacy. Tom was also involved with Cigital's early Java Security efforts, helping to co-author an appendix on Java code signing for the 1999 McGraw and Felten “Security Java” book. Prior to joining Pikewerks, Tom spent two years at Cyveillance working on open source intelligence applications. A main focus for Tom at Cyveillance was scanning the Internet for compromised credit card and social security numbers on web sites, FTP drop sites used by malware, and IRC channels used for the sale and exchange of stolen credentials. Tom also assisted in operating Cyveillance's monthly web crawl and index of over 100 million domains, helping to increase automation and predictability.

Tom's skill set includes development on Microsoft Windows and Linux platforms, in multiple languages such as C, C++, Java, and Python, and for multiple relational database systems such as Microsoft SQLServer, MySQL, and IBM DB2.

Kenneth Prole, Project Engineer at AVI/Secure Decisions Inc.

Proposed Role:	Research Lead
Company:	AVI-Secure Decisions
Proposed Level of Support	25%

Ken Prole is a Project Engineer at the Secure Decisions Division of Applied Visions, Inc. with extensive experience in visualization and information assurance applications. He has over twelve years of experience developing visualization solutions for both government and commercial clients. He is currently leading a DARPA funded SBIR project called MeerCAT, which visualizes wireless transmitters. This project is being transitioned into use by the DoD through DISA funding and was selected as a DARPA success story. Ken is also leading the visualization development for the DARPA sponsored National Cyber Range program. Prior to leading the these projects, Ken led large scale government research projects for DARPA and the Department of Homeland Security, applying his extensive knowledge in security visualization and information assurance to help protect the Department of Defense from cyber attacks. Mr. Prole holds a Master's degree from Long Island University, C.W. Post and a Bachelor's degree from Marist College, both in Information Systems. Ken holds a TS clearance and has a Patent Pending for Multilayer Wireless Network Flow Graph.

- Coauthored selected Publications include: "Advances in Topological Vulnerability Analysis," in *Proceedings of the Cybersecurity Applications & Technology Conference for Homeland Security 2009*; "Wireless Cyber Assets Discovery Visualization," in *VizSec 2008; and*, "A Graph-Theoretic Visualization Approach to Network Risk Analysis," *VizSec 2008*.

Phillip Porras, Program Director of Systems Security Research	
Proposed Role:	Research Area Lead
Company	SRI International
Proposed Level of Support:	25%
Education:	M.S. Computer Science
Location:	Menlo Park, California

Phillip Porras is a Program Director of systems security research in the Computer Science Laboratory at SRI International, and has been a Principal Investigator for many research projects sponsored by DARPA, DoD, NSF, NSA, and others. He is currently a Principal Investigator in a multi-organization NSF research project, entitled "Logic and Data Flow Extraction for Live and Informed Malware Execution." He leads a research project studying malware pandemics on next generation networks for the Office of Naval Research. He is also the Principal Investigator of a large ARO-sponsored research program entitled Cyber-TA, which is developing new techniques to gather and analyze large-scale malware threat intelligence across the Internet. Phillip's most recent research prototype technologies include BotHunter (http://www.bothunter.net), BLADE (ww.blade-defender.org), Highly Predictive Blacklists (http://www.cyber-ta.org/releases/HPB/), and the Eureka malware unpacking system (eureka.cyber-ta.org). He has been an active researcher, publishing and conducting technology development in intrusion detection, alarm correlation, malware analysis, active networks, and wireless security. Previously, he was a manager in the Trusted Computer Systems Department of the Aerospace Corporation, where he was also an experienced trusted product evaluator for NSA (which includes security testing, risk assessment, and penetration testing of systems and networks). Phillip has participated on numerous program committees and editorial boards, and on multiple commercial company technical advisory boards. He holds eight U.S. patents, and has been awarded Best Paper honors in 1995, 1999, and 2008.

Jason Upchurch, Senior Technical Lead for Intrusions Forensics	
Proposed Role:	Research Area Lead
Company	GDAIS
Proposed Level of Support:	25%
Education:	B.S. Computer Science, Regis University, 2007
Location:	Centennial, Colorado

Jason Upchurch has extensive experience as a technical manager and subject matter expert in malware analysis and intrusion forensics. He is currently a senior technical lead for GDAIS Cyber Systems. He is responsible for leading incident response and forensics relating to computer intrusions and reports to the Director of Cyber Systems. In addition, he provides mentoring/coaching to other cyber systems personnel, develops automation techniques for digital forensics, and provides training both internally and externally on Malware Analysis and Large Dataset Forensics. He has presented at conferences at the national and international level.

Jason was the technical lead and contract manager for both the Defense Computer Forensics Laboratory (DCFL) Intrusion Section, to include the malware analysis unit, and the contract personnel in the National Cyber Investigative Joint Task Force (NCIJTF) and the Defense Collaborative Investigative Environment (DCISE). He lead the effort for malware analysis development at the DoD Cyber Crime Center and was the center's first malware analyst. In these roles he was instrumental in guiding the process for malware analysis and cyber intelligence within the DoD LE/CI community. Jason has been conducting computer forensics professionally since 1999.

II.F Summary Slides

Cyber Physiology Analysis Framework

Concept

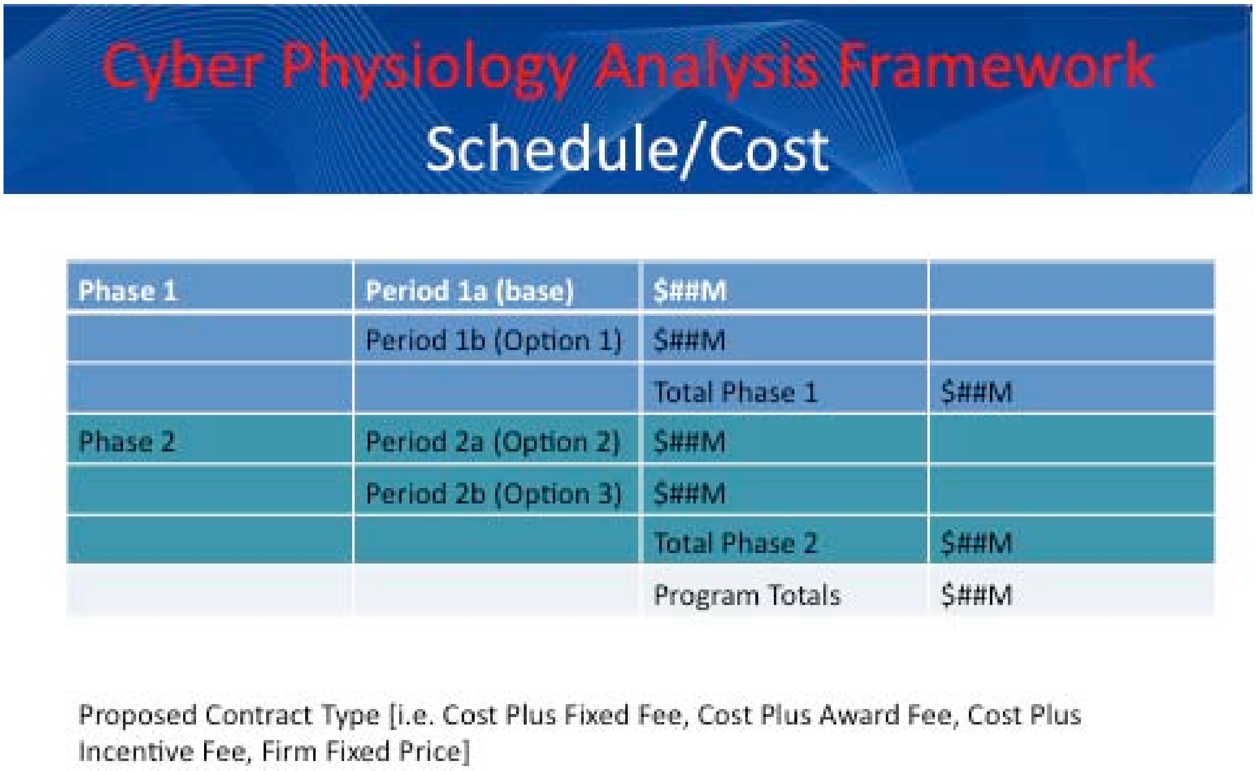
Create an operational framework for

II.F Summary Slides

Cyber Physiology Analysis Framework

Contract/Proposal Specifics

Intellectual Property
Data Rights Summary
Deliverables

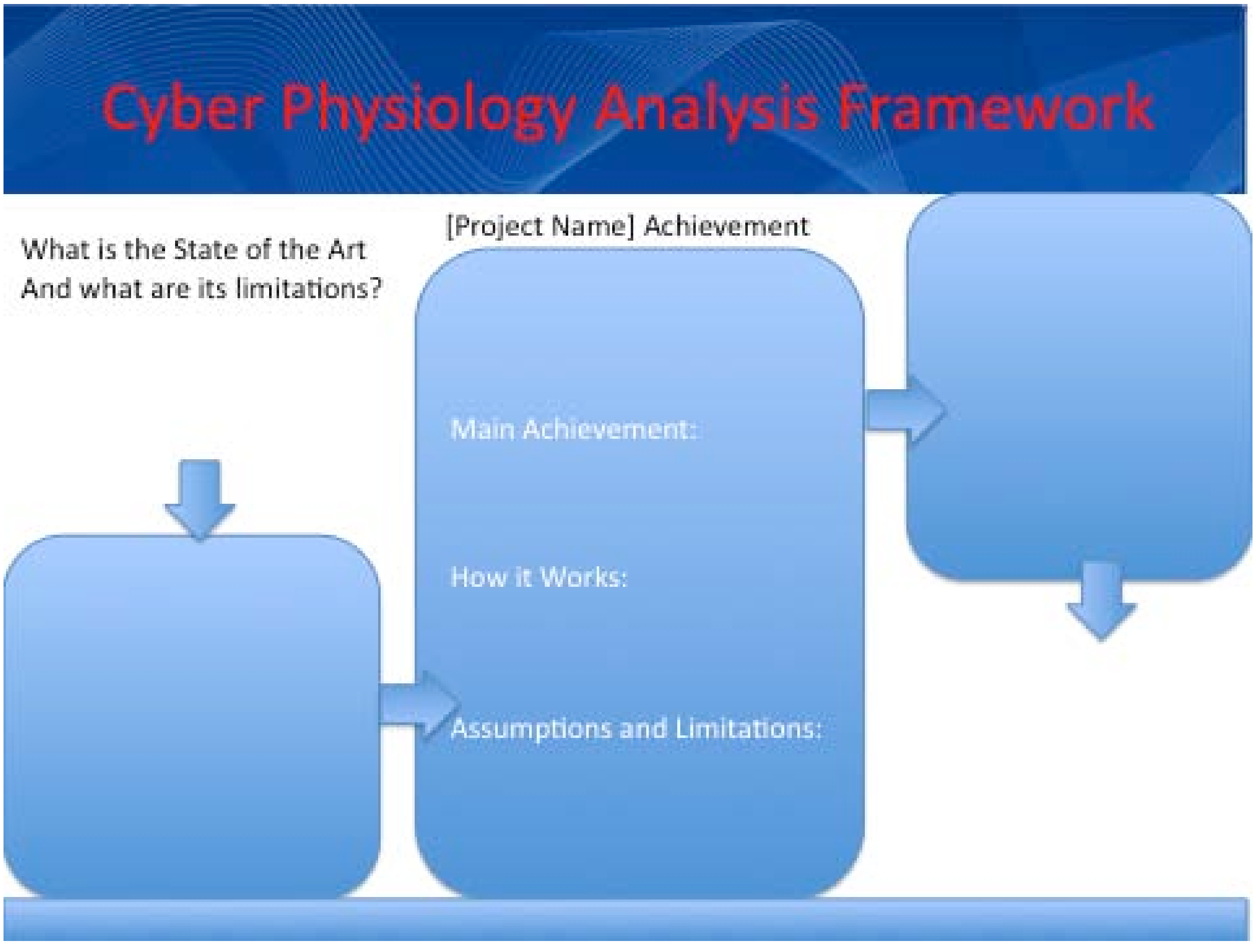


Cyber Physiology Analysis Framework

Schedule/Cost

Phase 1	**Period 1a (base)**	**$##M**	
	Period 1b (Option 1)	$##M	
		Total Phase 1	$##M
Phase 2	Period 2a (Option 2)	$##M	
	Period 2b (Option 3)	$##M	
		Total Phase 2	$##M
		Program Totals	$##M

Proposed Contract Type [i.e. Cost Plus Fixed Fee, Cost Plus Award Fee, Cost Plus Incentive Fee, Firm Fixed Price]



Cyber Physiology Analysis Framework

What is the State of the Art
And what are its limitations?

[Project Name] Achievement

Main Achievement:

How it Works:

Assumptions and Limitations:

Section III. Detailed Proposal Information

III.A Statement of Work (SOW)

The HBGary Federal Team will execute the Statement of Work in accordance with the Work Breakdown Structure (WBS) developed for the DARPA Cyber Genome (DCG) Program, consisting of the following seven major Tasks: Task 1 – Specimen Feeds and Pre-processor; Task 2 - Specimen Repository; Task 3 - Specimen Analysis & Visualization Interface; Task 4 - Genomes Library; Task 5 - Traits Library; Task 6 - Static Memory Analysis and Runtime Tracing; Task 7 - Belief Reasoning and Inference Network.

III.A.1 Program Management

The HBGary Federal Team will use suitable program and subcontract management practices to attain the technical, cost and schedule goals of the DCG program. We conduct internal technical interchange meetings to facilitate performance on our programs, with quarterly program reviews and a final review with DARPA at the conclusion of each phase. Quarterly reviews will be held at different contractor locations, or with DARPA's concurrence, at other facilities to permit demonstrations of incremental system capabilities. The HBGary Federal team will divide the work according to our strongest competencies and adjust work share appropriately as the research progresses.

Date	Description	Type
Monthly	Financial Reports	Document
NLT 30 days EOP	Technical and Financial Plan/Report	Document
NLT EOP	Software Documentation (Design, Instructions, Use)	Document
NLT 3 days EOP	Annual Review	Presentation
EOP	Final Report	Document

III.A.2 SOW Tasks

III.A.2.1 Task 1: Specimen Feeds & Pre-Processor: SRI Lead

Team Member SRI shall provide research and development of techniques for unpacking and de-obfuscating malware, as well as identification and remediation of malware trigger and anti-analysis techniques. This includes developing and refining research papers and prototypes for each of these capabilities.

Team Member Pikewerks shall provide research and development of Linux malware capture capabilities including next generation honeynets, client-side malware, email-borne malware, and malware embedded in p2p networks. This will include support for the development of novel and scalable automated unpacking/de-obfuscation techniques for captured malware.

Table 1. Task 1 – Detailed Task Description and Duration

Date	Effort	Performer
Months 1-12	Establish basis of research for automated unpacking/de-obfuscation of malware.	SRI
Months 1-12	Establish basis of research for identifying malicious logic and anti-analysis techniques in malware	SRI
Months 12-24	Develop a prototype for automated unpacking/de-obfuscation of a subset of packing/obfuscation techniques.	SRI
Months 12-24	Research methodologies for automated remediation of malicious logic and anti-analysis techniques.	SRI
Months 24-36	Refine techniques and prototype for automated unpacking/de-obfuscation.	SRI
Months 24-36	Develop a prototype of automated remediation of malicious logic and anti-analysis techniques	SRI
Months 36-48	Refine automated remediation of malicious logic and anti-analysis prototype	SRI
Months 1-6	Establish basis of research, proof of concept and methodologies for acquiring Linux-based malware with an emphasis on current specimens.	Pikewerks
Months 6-12	Develop prototype(s) for acquiring Linux-based malware	Pikewerks
Months 1-12	Provide support in research and development of automated unpacking/de-obfuscation techniques for Linux-based malware	Pikewerks
Months 12-24	Provide support in research and development of automated unpacking/de-obfuscation techniques for Linux-based malware	Pikewerks
Months 12-24	Mature prototype capabilities to acquire Linux-based malware in the wild.	Pikewerks
Months 24-36	Maintain acquisition capability of new Linux-based malware through development of new techniques (honeypots, clients, etc).	Pikewerks
Months 36-48	Maintain acquisition capability of new Linux-based malware through development of new techniques (honeypots, clients, etc).	Pikewerks

Table 2. Task 1 – WBS Milestones, Completion Criteria and Deliverables

Planned Date	**Milestones, Completion Criteria and Deliverables**	Performer
Month 12	Deliver research paper and proof of concept for automated unpacking/de-obfuscation of binaries and code not mapped to process memory	SRI
Month 12	Deliver a research paper on malicious logic and anti-analysis techniques.	SRI
Month 24	Deliver updated research paper on refined unpacking/de-obfuscation techniques and deliver prototype to cover a subset of high priority/high volume packing/obfuscation technologies.	SRI
Month 24	Deliver a proof of concept and research paper on removal of malicious logic and anti-analysis techniques	SRI
Month 36	Deliver an enhanced prototype for automated de-obfuscation/unpacking of a larger subset of malware packing/obfuscation techniques	SRI

Month 36	Deliver a full-features prototype and demonstration on malicious logic and anti-analysis techniques with updated research paper.	SRI
Month 48	Deliver a fully automated prototype for removal of malicious logic and anti-analysis techniques with updated research paper.	SRI
Month 2	Deliver Linux-based malware feeds or specimens necessary for the project.	Pikewerks
Month 6	Deliver research paper and proof of concept for methods to acquire current Linux-based malware specimens (i.e. honeynets, client capture, email, document, or p2p embedded.	Pikewerks
Month 12		Pikewerks
Month 24		Pikewerks

Task 1 Dependencies

Task 1 activities are not dependant on other DCG Tasks..

III.A.2.2 Task 2: Specimen Repository: HBGary Federal Lead

HBGary Federal will develop a specimen repository, which will be used to store live malware samples and their associated metadata.

Table 3. Task 2 - Detailed Task Description and Duration

Date	Effort	Performer
Months 1-3	Develop database schema for storing malware samples and their associated metadata. Design architecture to host the Specimen Repository,	HBGary Federal
Months 3-4	Implement Specimen Repository Database and configure architecture.	HBGary Federal
Months 4-12	Refine database schema to incorporate new knowledge gained through research on other DCG tasks.	HBGary Federal

Table 4. Task 2 - Milestones, Completion Criteria and Deliverables

Planned Date	**Milestones, Completion Criteria and Deliverables**	Performer
Month 3	Deliver database design document for Specimen Repository.	HBGary Federal
Month 4	Deliver Specimen Repository software architecture.	HBGary Federal
Month 12	Deliver refined Specimen Repository software architecture.	HBGary Federal

Task 2 Dependencies

Task 2 activities are dependant upon obtaining sample of malware specimens collected during Task 1.

III.A.2.3 Task 3: Specimen Analysis & Visualization Interface: AVI/Secure Decisions Lead

Team MemberAVI/Secure Decisions, supported by GDAIS, will develop visual tools to support the visual representations of malware traits, sequences, and physiology profiles. These will aid analysts in the identification of new traits, genomes, and aggregate malware types and unique compositions, and assist in the understanding of malware's overall function, behavior and intent through these visual cues.

Table 5. Task 3 - Detailed Task Description and Duration

Date	Effort	Performer
Months 1-6	Define visualization requirements for the analysis of malware functionality and behaviors.	AVI/Secure Decisions
Months 7-8	Describe and document an architecture that visualizes malware functionality and behaviors	AVI/Secure Decisions
Months 9-12	Develop visualization prototypes to assist in the analysis of malware functionality and behaviors.	AVI/Secure Decisions
Months 12-24	Integrate and demonstrate progressively more complete visualization prototypes	AVI/Secure Decisions
Months 19-21	Define requirements for the visualization of aggregate malware functionality and behaviors (fingerprinting and auto-discovery of characteristics through visual cues.	AVI/Secure Decisions
Months 22-23	Describe and document an architecture that visualizes aggregate malware functionality and behaviors (fingerprinting and auto-discovery of characteristics through visual cues.	AVI/Secure Decisions
Months 1-12	Provide malware analysis expertise and operational relevance to the developed analysis interfaces and products developed in phase 1a	GD AIS
Months 12-24	Provide malware analysis expertise and operational relevance to the developed analysis interfaces and products developed in phase 1b	GD AIS

Table 6. Task 3 - Milestones, Completion Criteria and Deliverables

Planned Date	**Milestones, Completion Criteria and Deliverables**	Performer
Month 6	Deliver research paper on visualization for analysis of malware behavior and functions.	AVI/Secure Decisions
Month 8	Deliver research paper on visualization architecture and proof of concept for malware functions and behaviors.	AVI/Secure Decisions
Month 12	Deliver prototype capability for the visualization of malware functionality and behaviors	AVI/Secure Decisions
Month 24	Deliver enhanced prototype with fully functional capability to visualize malware functionality and behaviors.	AVI/Secure Decisions
Month 21	Deliver a research paper on the visualization of aggregate malware functionality and behaviors, including the ability to identify and classify malware based on its visual cues.	AVI/Secure Decisions
Month 23	Deliver research paper on visualization architecture and proof of concept of malware aggregate functionality and behaviors.	AVI/Secure Decisions

Task 3 Dependancies

Task 3 activities are dependant upon the outputs of Tasks 4,5, and 6.

III.A.2.4 Task 4: Genomes Library: HBGary Federal Lead

HBGary Federal will provide research and development of complex, clustered, or sequenced functions and behaviors (genomes) to fully enumerate and qualify overall malware functions, behavior, and intent.

Table 7. Task 4 - Detailed Task Description and Duration

Date	Effort	Performer
Months 12-24	Establish basis of research for identification and mathematical representation of Windows-based malware complex, clustered, or sequenced functions (genomes).	HBGary Federal
Months 24-36	Research and develop Windows base genome datasets of linear execution space.	HBGary Federal
Months 36-48	Research and develop more sophisticated Windows genome datasets in linear execution space.	HBGary Federal
Months 12-48	Provide support to Windows based Genome datasets.	HBGary
Months 12-24	Establish basis of research for identification and mathematical representation of linux-based malware complex, clustered, or sequenced functions (genomes).	Pikewerks
Months 24-36	Research and develop base genome datasets of linear execution space.	Pikewerks
Months 36-48	Research and develop more sophisticated genome datasets in linear execution space.	Pikewerks

Table 8. Task 4 - Milestones, Completion Criteria and Deliverables

Planned Date	Milestone	Performer
Month 24	Deliver research paper and proof of concept for enumerating higher level complex behaviors and functions (genomes) of Windows-based malware, including techniques and mathematical models used.	HBGary Federal
Month 36	Deliver Windows genomes library	HBGary Federal
Month 48	Deliver a more extensive Windows genomes library	HBGary Federal
Month 24	Deliver research paper and proof of concept for enumerating higher level complex behaviors and functions (genomes) of linux-based malware, including techniques and mathematical models used.	Pikewerks
Month 36	Deliver genomes library	Pikewerks
Month 48	Deliver a more extensive genomes library	Pikewerks

Task 4 Dependencies

Task 4 Genome Library activities are dependant upon Task 5 Traits Library and the output of Task 6.

III.A.2.5 Task 5: Traits Library: HBGary Federal Lead

HBGary Federal will conduct research and develop a malware traits library for the purposes of identifying and qualifying malware discrete functions and behaviors that will be used as the building blocks for evaluating malware function, behavior, and intent. This will include research and development of toolmarks and latent artifacts within linux executables that can reveal information about the environment when developed and compiled.

Table 9. Task 5 - Detailed Task Description and Duration

Date	Effort	Performer
Months 1-12	Establish basis of research for identification and mathematical representation of Windows-based malware behavior and function (traits).	HBGary Federal
Months 12-24	Research and develop simple traits datasets of Windows linear execution space.	HBGary Federal
Months 24-36	Research and develop complex traits datasets of Windows linear execution space.	HBGary Federal
Months 1-36	Provide support to Windows based Trait development.	HBGary, Inc.
Months 1-12	Establish basis of research for identification and mathematical representation of linux-based malware behavior and function (traits).	Pikewerks
Months 12-24	Research and develop simple traits datasets of linear execution space.	Pikewerks
Months 24-36	Research and develop complex traits datasets of linear execution space.	Pikewerks
Months 1-48	Provide 400 hours of support to HBGary Federal in the development of malware traits.	GD AIS

Table 10. Task 5 - Milestones, Completion Criteria and Deliverables

Planned Date	**Milestones, Completion Criteria and Deliverables**	Performer
Month 12	Deliver research paper on methodology for Windows-malware function enumeration including mathematical language and models used to qualify traits	HBGary Federal
Month 24	Deliver foundational Windows traits library	HBGary Federal
Month 36	Deliver complex Windows traits library	HBGary Federal
Month 12	Deliver research paper on methodology for Linux-malware function enumeration including mathematical language and models used to qualify traits	Pikewerks
Month 24	Deliver foundational traits library	Pikewerks
Month 36	Deliver complex traits library	Pikewerks

Task 5 Dependencies

Task 5 activities are dependant upon Task 6.

III.A.2.6 Task 6: Static Memory Analysis & Runtime Tracing: HBGary Inc. Lead

HBGary will conduct research and develop automated methods to exercising Linux-based malware full execution paths for the purposes of providing a complete analysis of malware behavior, functionality, and intent.

Table 11. Task 6 - Detailed Task Description and Duration

Date	Effort	Performer
Months 12-24	Establish basis of Windows research and methodology for using static and dynamic analysis to discern variables required for greater function tree execution	HBGary
Months 24-36	Develop a Windows proof-of-concept capability to automatically identify and exercise variables to achieve greater branch execution coverage	HBGary
Months 36-48	Develop an enhanced prototype capability to automatically identify and exercise variables to achieve greater branch execution coverage	HBGary
Months 12-24	Establish basis of Linux research and methodology for using static and dynamic analysis to discern variables required for greater function tree execution	Pikewerks
Months 24-36	Develop a Linux proof-of-concept capability to automatically identify and exercise variables to achieve greater branch execution coverage	Pikewerks
Months 36-48	Develop an enhanced prototype capability to automatically identify and exercise variables to achieve greater branch execution coverage	Pikewerks

Table 12. Task 6 - Milestones, Completion Criteria and Deliverables

Planned Date	**Milestones, Completion Criteria and Deliverables**	Performer
Month 24	Deliver research paper and Windows proof of concept for using static and dynamic analysis to discern variables required for greater function tree execution.	HBGary
Month 36	Deliver a Windows prototype capability to automatically identify and exercise variables to achieve greater branch execution coverage	HBGary
Month 48		HBGary
Month 24	Deliver research paper and Linux proof of concept for using static and dynamic analysis to discern variables required for greater function tree execution.	Pikewerks
Month 36	Deliver a Linux prototype capability to automatically identify and exercise variables to achieve greater branch execution coverage	Pikewerks
Month 48		Pikewerks

Task 6 Dependencies

Task 6 activities are not dependant on other DCG Tasks.

III.A.2.7 Task 7: Bayesian Reasoning & Inference Network: HBGary Federal Lead

HBGary Federal will conduct research and develop a belief network model that can be trained and used to classify a malware object into categories. This will require processing a large set of known malware and a large set of known “clean” applications and code so that the model can reliably judge the intent of a given binary. A stochastic approach, such as a Belief inference model, can be matched with the probabilities learned and weights given to individual traits and behaviors.

Table 13. Task 7 - Detailed Task Description and Duration

Date	Effort	Performer
Months 24-36	Perform research, design and proof of concept development.	HBGary Federal
Months 36-48	Develop proof-of-concept of belief reasoning capability.	HBGary Federal

Table 14. Task 7 - Milestones, Completion Criteria and Deliverables

Planned Date	Milestones, Completion Criteria and Deliverables	Performer
Month 36	Deliver research paper, design document and proof of concept demonstration.	HBGary Federal
Month 48	Deliver demonstration of proof of concept belief reasoning capability.	HBGary Federal

Task 7 Dependancies

Task 7 activities are dependant upon Task 4, 5, and 6.

III.B Description of the Results

A successful cyberdefense tool must not only offer the needed technical capabilities to identify and isolate malware, but also offer the integration, utility and support users expect from commercial tools. HBGary and Pikewerks have track records of commercialization success. We know the difficulties in technology transition and commercialization. Software won't transition very far in government or to the public if it is not of commercial grade. Our team knows from experience that it costs considerably more money and effort to develop commercial grade, production software than R&D prototypes. Quality software that meets customer needs doesn't ensure success alone. Senior marketing and sales personnel with proven track records are needed to take new products to market. Effective marketing requires messaging that resonates with paying customers, sales collateral tools, full feature website, trade show presence, conference speaking, case studies, press releases, press interviews, and strategic alliances. After the sale customers need training classes and ongoing software maintenance and tech support. Furthermore, strategic commercialization alliances with larger companies are critical to success. Our team has already begun to discuss eventually co-licensing and reselling technologies developed as part of this Cyber Genome Program.

III.C Detailed Technical Rationale

The HBGary Federal Team has tremendous experience with leading malware analysis methods, techniques, and capabilities to draw from to develop successful approaches to the challenges of the cyber genome project. We will make advances in several state-of-the-art capabilities to create an automated malware system that will discern good from bad behavior, classify the myriad of possible functions in software, and determine a specimen's overall capabilities and purpose.

The first challenge to be addressed is the best method for reliably extracting content from a given specimen for analysis. There are a few approaches:

- Static Binary Analysis. This is the traditional method of analyzing malware. It relies upon tools like IDA Pro and a strong library of specialized tools to unpack/de-obfuscate code to get to analyzable data. One of the largest negatives for this method is that code packers/obfuscators are usually a step ahead of the unpackers/de-obfuscators. Another negative is that self-modifying code can be very difficult to analyze.
- Static Memory Analysis. Image physical memory followed by automated reconstruction of the image

including the operating system, all running programs and overall state of the computer. It is possible that malware could detect memory imaging is occurring then giving back false information to hide its existence (but we have seen no evidence of any malware doing this). Once memory is successfully imaged, there is no thwarting memory analysis.

- Runtime Analysis. Involves executing the specimen in a controlled, instrumented, typically virtual environment, and recording all of the API calls, registry entries, etc. This requires a system that avoids detection by the binary (anti-debugging tricks). Runtime analysis is limited to recording behaviors that a binary exhibits in a small window of time. A large negative is that many potential behaviors are never called or executed in a binary until specifically requested by an attacker. A negative is that complete discovery of all code paths may be an intractable problem, either requiring too much processing power or too much memory/space to solve in a reasonable time frame. A positive is that we don't have to worry about packers and obfuscation, but we do have to prevent the binary from detecting that it is in a controlled environment. Additionally, this approach allows for integrating different tools to probe or test malware, making the overall system more extendable.

We assert the best specimen recording approach involves a combination of all three methods, mixing the information gained from static file and memory analysis with a run-time execution system. This approach will allow us to identify and mitigate anti-analysis and security techniques, get a true representation of the program while executing, and recover a more significant amount of code paths.

We have selected a trait (gene) and pattern (genome) approach to discern malware functionality and behavior because we believe this gives us maximum flexibility in evolving the system as well as the highest level of fidelity of the components of the specimen. In many cases the traits themselves will likely be neutral, however the patterns and context exhibited will display malicious or benign behaviors. This approach allows us to evolve the traits and patterns independently and to more dynamically mature trait and pattern libraries. This approach should also provide benefit to evolution and lineage. We have experience and capability using this approach to satisfy more simplified goals of malware detection that are very successful.

Lastly to reach the goal of true automation you need a system that can learn from existing models and determine functionality and behavior of future unidentified malware and its traits and patterns. Fitting within the overall approach, we believe a Belief Reasoning Engine, like Dempster-Shafer, to be the most appropriate solution to be developed for this area.

III.D Detailed Technical Approach

We believe the best approach is to start by researching the detailed mechanisms of software and develop a language and ruleset that accurately qualifies discrete software functions and behaviors, followed by an aggregate analysis of discrete functions to discern patterns; sequences and clusters of these traits that connote a higher order of software functionality and behaviors. Part of our research will focus on best methods to exercise software in an analysis environment to expand our visibility into variable dependent branches in code. The research will be tied together through a reasoning engine that can make automatic probability decisions on the behavior and functionality of malware based on historical inference models. The final goal will be to submit an unknown malware specimen with previously undocumented functions and behaviors and automatically generate a cyber physiology profile that characterizes the new traits and discerns and describes the overall function, behavior, and intent of the malware with an easily digestible visual format. This format we are calling the Cyber Physiology Profile that will represent both the mathematical, visual, and descriptive characterizations of the specimen.

III.D.1 Specimen Collection and Pre-Processing

Collection methods need to be addressed to ensure we are developing capabilities using the most recent and challenging malware specimens available. There are feeds for malware to which we have existing subscriptions and will research to ensure we have the most relevant data available. In addition we will conduct research and develop malware harvesters and honeynets to collect malware in the wild not contained in feeds. The challenge here is in finding or attracting malware that has propagated under the radar enough so as not to have been detected and collected by one of the feed providers. Variations of honeypots have been in existence for many years on both windows and Linux platforms. Where our research differs is in an integrated approach between collection and analysis that trains our sensors how to behave in order to maximize new collections.

We propose to research and develop a passive and active collection capability for Linux and Windows-based malware using virtualized clients and webhosts configured with variations of operating systems, patches, and services. The passive systems will emulate persistent, commercial web services, while the active systems will emulate client systems that will browse websites, conduct p2p file transfers, open email attachments, and perform numerous other high-risk activities. The personas of the passive and active systems will receive periodic updates through scripts that pull from the malware repository ensuring maximum exposure to new collections.

Increasingly malware employs sophisticated anti-detection and analysis techniques such as; obfuscation, packing, encryption, and modularization. While conducting malware analysis on running programs alleviates some of the complexity since binaries to run typically need to be complete, unpacked, and unencrypted, their are exceptions and there are techniques used by malware authors to try and protect malware from analysis. The goal of the research in this phase is to investigate methods used to protect malware from detection and analysis and develop capabilities that allow automated analysis to continue.

We propose to research and develop binary evaluation metrics for the purpose of assessing the quality of the unpacked code. The post unpacking analysis capability will be delivered as an add-on to the Eureka framework to enable further analysis and classification of malware and will integrate SRI's speculative API resolution algorithm to automatically resolve call sites. We will develop additional criteria that determine the optimal moment for taking a memory snapshot of the running process and recovering the original entry point. We will also investigate novel ways of hiding Eureka from being detected by the running binary to avoid triggering suicide logic and explore snapshot-stitching techniques for dealing with multi-stage packers and block encryption.

As the origin entry point of windows based malware binary is usually not known at the point of unpacking, we will explore and implement novel strategies to uncover the OEP in the captured memory image of the process. We will then automatically rewrite the binary's header to set the OEP, rebuild import tables and research automated techniques for informed reconstruction of malware binaries to enable execution in a manner that bypasses environment checks and suicide logic. The output from static analysis of malware samples will enable guided executions of unpacked binaries.

Lastly, we will research and develop automated ways to recognize obfuscated code, identify various obfuscation steps employed to hinder automated analysis, and systematically employ de-obfuscation to restore the binary to an equivalent but un-obfuscated form. This will inspire new research and development of advanced and automated binary rewriting techniques.

III.D.2 Specimen Repository

Each of the phases within the cyber physiology analysis framework collects, analyzes, and outputs some form of data. It is the data output from each of these phases that interconnects within the rest of the framework. This being the case the Specimen Repository, while not an advanced area of research, plays a critical role within the overall effort. The various types of data that will need to be stored include; raw malware objects, specimen externals metadata, memory snapshot metadata, runtime data, cyber physiology profile data. We will develop mechanisms to check for duplications as well as updates to previously archived specimen.

Our database implementation will utilize both the database as a central repository for the data collected from the varying applications and the file system for storing compressed versions of the specimens. We will also normalize the data stored within the database to provide a system that will eliminate duplicate data, provide faster access to the available data, as well as provide a means for comparisons and versioning to calculate possible updates to specimens within the repository.

III.D.3 Specimen Analysis and Visualization Interface (SAVI)

Even in an automated malware analysis system there needs to be a human interface to aid in training the system, verifying data, and viewing results. Today most malware analysis is still a slow and tedious process that requires highly trained and frequently unavailable reverse engineers and malware analysts to do the work. Even tools such as those developed by the HBGary Federal team that expedite the reverse engineering process and display information in far more digestible forms stop short of displaying more simplified visual representations of malware that show at a glance the characteristics of a malware specimen.

We propose to research and develop a Specimen Analysis and Visualization Interface (SAVI), investigating various representations of malware that can provide information at a glance to the analysts, and allow the analyst to visualize malware in different ways from an aggregate view drilling down to a more interactive detailed view. The displays will be interactive in the sense that the analyst will be able to flag code segments, functions within the graphical view and pull up a more traditional analyst view for further inspection, make modifications, then revert to the graphical view to see how the changes affected the overall specimen representation.

Malware analysis based on multiple dimensions, and collection methods can lead to copious amounts of data that needs to be presented to the operator. We propose to visually represent this copious data using **multiple coordinated views, starting out with a high level overview, and then providing details-on-demand**. Figure #, is an example of a Secure Decision's developed visualization tool to represent running code. In our approach we will provide the user with an interface that guides the analyst's analysis and discovery of traits and patterns.

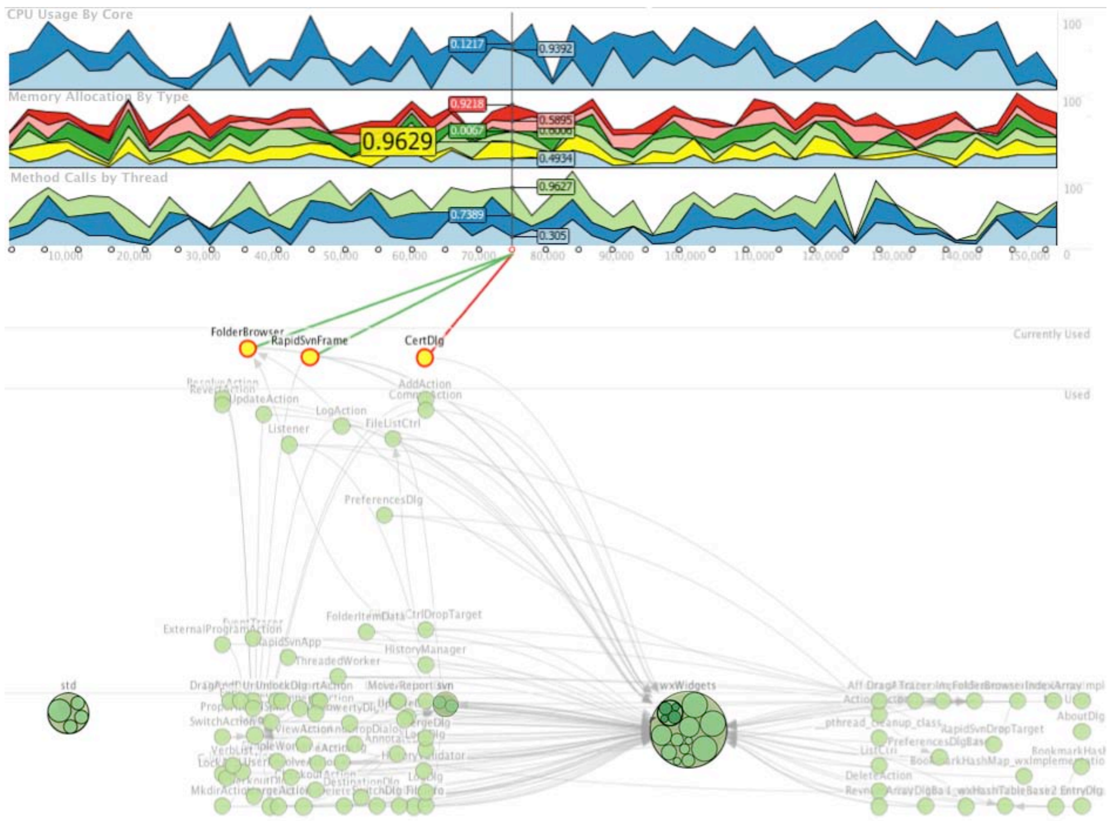


Figure #. Screenshot showing the contextual information of a running code (top) lined with the software structure information (bottom)

We will develop **prototype visualizations** based on factors such as exhibited traits, external and environmental artifacts, space and temporal artifact relationships, sequencing. This will support the identification and understanding of functions and behaviors to aid malware analysts in developing new traits and patterns of significance. They will also develop visual representations of a **Malware's Physiology Profile** to provide visual fingerprinting capabilities to malware analysts and to provide graphical cues for physiology reports. Figure #, is an example of a Secure Decisions developed visualization showing class dependencies in software.

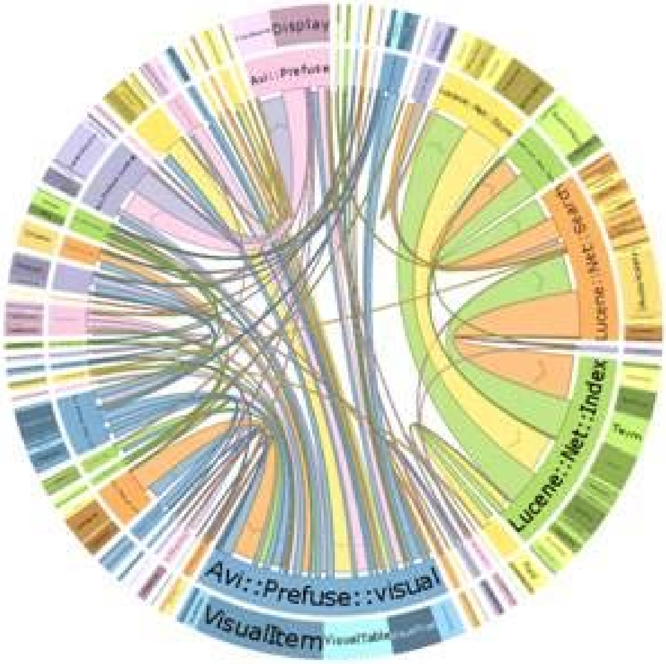


Figure #. iTVO screenshot showing dependencies between classes

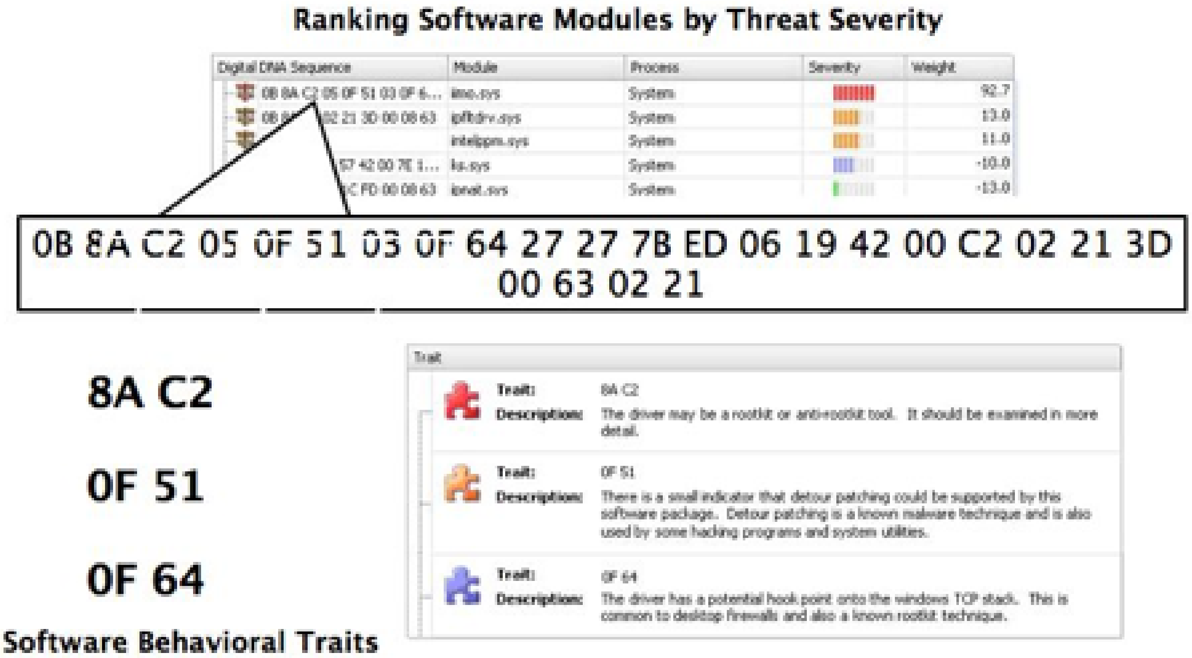
This type of representation of traits, patterns, and other internal artifacts would bring efficiency to the malware analysis process. Secure Decisions has an extensive visualization toolkit that can be leveraged to create novel visualization for malware analysis. Our tools and skills have been used to prototype and field a variety of visualizations for government and commercial cyber defense experts.

III.D.4 Traits Library

At its most fundamental level malware objects are a compilation of discrete functions that do work. In order to build a capability to automatically analyze malware for aggregate function and behavior we believe you must first accurately qualify all of its discrete parts. We propose to build a body of knowledge about code (aka, Traits), for example:

1. Identify Usage of API or system calls (WriteFile, RegOpenKey, InternetConnect, libc functions in Linux, etc.)
2. Identify algorithms in code logic (copy loop, decrypt block, parse string, etc)
3. Identify typical coding structures such as (if/else blocks, do/while loops, class structures, etc)

We propose to research and develop a trait coding system, an example of which is HBGary's existing trait coding system used to detect the presence of malware, as shown in Fig. #. The existing trait system is compr ised of the rules, an expression language, and a fuzzy matching system. We will use the existing system as a basis of research to determine the best methodology for developing a more complete trait coding system for the purposes of enumerating the low level and high level functions and behaviors for a more sophisticated analysis of the malware specimen.


Ranking Software Modules by Threat Severity
0B 8A C2 05 0F 51 03 0F 64 27 27 7B ED 06 19 42 00 C2 02 21 3D
00 63 02 21
8A C2
0F 51
0F 64
Software Behavioral Traits

Digital DNA Sequence	Module	Process	Severity	Weight
0B 8A C2 05 0F 51 03 0F 6…	ilmo.sys	System	▮▮▮▮▮▮▮	92.7
0B 8[illegible] 02 21 3D 00 08 63	ipfltdrv.sys	System	▮▮▮▮	13.0
	intelppm.sys	System	▮▮▮▮	11.0
57 42 00 7E 1…	ks.sys	System	▮▮▮	-10.0
C FD 00 08 63	ipnat.sys	System	▮	-13.0

Trait

Trait: 8A C2
Description: The driver may be a rootkit or anti-rootkit tool. It should be examined in more detail.

Trait: 0F 51
Description: There is a small indicator that detour patching could be supported by this software package. Detour patching is a known malware technique and is also used by some hacking programs and system utilities.

Trait: 0F 64
Description: The driver has a potential hook point onto the windows TCP stack. This is common to desktop firewalls and also a known rootkit technique.

Figure x: HBGary's Trait Coding System for Detecting Malware

III.D.5 Genomes Library

Using the traits library we will research and develop a patterns or genomes library. While some traits alone can aid in the detection or identification of potentially malicious activity in code, such as specimen uses a packer, the traits alone are not enough to determine automatically the aggregate functions and behaviors of a specimen. For example, some malware might try to elevate privileges, or open up a file and directly after open a network connection, or try to use obfuscation techniques. In each of these cases there are legitimate programs, even security programs, which would employ these functions or exhibit this type of behavior. So with traits alone the best you might be able to develop is a probability based on an aggregate of traits exhibited.

To truly develop a comprehensive view of malware behavior and function takes some analysis of the traits and the patterns they exhibit in malware. As an example, noticing the following traits in a code sequence: URLDownloadToFile(somefile.exe) followed by CreateProcess(somefile.exe). This could be labeled as a "Download and execute" pattern, and the intent could be identified as "Suspicious", or the behavior as "Risky" or "Dangerous". We propose to research and develop patterns of traits, such as sequencing or clustering, of good and bad software, to develop strong indicators that can be relied upon during automated analysis. In the case of sequence patterns, all of the traits need to fall into a particular sequence to flag as true, whereas with a cluster or grouping patterns they just have to occur in total or occur within certain proximity of each other. A third example would be patterns that occur within the presence of certain variables.

One model might be to apply the use of the patterns within specific genomes. So the first genome applied might be a classifier genome. The system would use weight values to determine if a program is malware. Once something has been determined as malware, it should be fed into a second genome. The second genome has trait-codes for all the code idioms used to develop software functions. For example, it would contain traits for all the ways a developer might code a TCP/IP recv loop. It would also contain all the trait patterns for malicious behaviors; such as all the ways a developer might sniff keystrokes. We could call this the lineage genome.

Finally, using the results from the lineage genome, analysts can develop archetypes, building statistical tools and visualization so that 'colonies' of largely similar malware can be grouped. When a new colony starts to form in the data-set, we can construct a new archetype to represent it. The archetype will contain the traits from the lineage genome that are common to most of the colony. Once the archetype has been created, malware can be automatically classified into the archetype as it comes in. The archetypes are not a genome, but a secondary layer of sorting over the lineage genome. This system should be able to predict upcoming attacks. When new samples are collected from the wild, they will automatically be classified into an archetype. A sudden growth of a new colony would represent a new malware variant that needs to be addressed. Any such outbreak would soon find a way into DoD and customer networks, so this offers a predictive capability for defense.

Finally, using the results from the lineage genome, analysts can develop archetypes, building statistical tools and visualization so that 'colonies' of largely similar malware can be grouped. When a new colony starts to form in the data-set, we can construct a new archetype to represent it. The archetype will contain the traits from the lineage genome that are common to most of the colony. Once the archetype has been created, malware can be automatically classified into the archetype as it comes in. The archetypes are not a genome, but a secondary layer of sorting over the lineage genome. This system should be able to predict upcoming attacks. When new samples are collected from the wild, they will automatically be classified into an archetype. A sudden growth of a new colony would represent a new malware variant that needs to be addressed. Any such outbreak would soon find a way into DoD and customer networks, so this offers a predictive capability for defense.

III.D.6 Static Memory Analysis and Runtime Tracing (SMART)

The SMART system will provide a nearly complete picture of the execution of any piece of software by combining the data acquired from three primary technologies:

- Runtime tracer
- Physical memory imaging and reconstruction
- Dataflow tracer

Runtime Tracer

The Runtime Tracer is a software tracing system and instrumented data collector capable of sampling and capturing data while tracing every process and every thread, both usermode and kernel mode, system wide and in real time. It will capture control and data flow at a single step resolution. Data sampling captures the contents of registers, the stack, and target buffers of de-referenceable pointers. Symbols are resolved for all known API calls, and when combined with argument sampling, will drastically reduce the time required to gain program understanding.

The Runtime Tracer's post-execution debugging is a paradigm shift from traditional interactive live debugging. While traditional interactive debugging is useful for software development, it is cumbersome when used for tracing program behavior. Traditional debugging tools are designed for control of software execution, as opposed to observation only. The reverse engineer only needs to *observe* the binary's behavior and data. The software under test is recorded during runtime. The analysis takes place later. Unlike traditional debuggers, the Runtime Tracer can follow multiple processes and trace parent/child process execution. It can also follow a process injecting a DLL into another process.

The Runtime Tracer operates at a very low level within the system, layering itself directly above the Hardware Abstraction Layer (HAL) and underneath the Windows kernel to provide complete control over the operating environment while at the same time maintaining performance levels to trace software in real time. It will not be bound by dependency on the Windows userland Debugging API and therefore will not be thwarted by malware anti-debugging tricks. The target software is not modified in any way. No breakpoints are injected. No thread context is changed. No debugger is attached. Tracing is performed completely external to the process operating environment.

Physical Memory Imaging and Reconstruction

Once the Runtime Tracer completes its runtime data collection, additional low level data can be harvested from physical memory. SMART will image physical memory (including RAM and pagefile) and reconstruct the operating system to recover all digital objects present in memory at the time of the image snapshot. Low level data collected will include executables, processes, drivers, modules, strings, symbols, network sockets, open files and data buffers. Any digital object can extracted, disassembled and examined down to its hexadecimal representation in memory. Because all objects and data are recovered they can also be inspected in relation to each other for contextual information.

Dataflow Tracer

To more fully understand a binary's functions and behaviors a skilled reverse engineer will "follow the data" to understand what code blocks operate on it and how. The engineer must emulate or model a computer system in his mind and keep painstakingly detailed and exhaustive notes of ever changing buffer values and data states. This work can take days or weeks depending on the program's size and how deeply he seeks to understand its behaviors.

We propose to develop an automated Dataflow Tracer to reveal complex relationships between code blocks and data which will take us far beyond low level data collection from runtime tracing and physical memory reconstruction. Dataflow tracing will associate different code blocks with each other by cross referencing common data and data derivatives used by code. Suppose code section A uses some data in memory and at a later time code section B uses the same data. It is very common that code blocks A and B can exist in different modules or threads while not appearing to be related in the code logic, but the fact that they operate on the same data establishes a relationship. Let's look at a simple example. Suppose a binary reads in an encrypted configuration file then later on other code decrypts it to reveal an IP address, which is copied and moved to another location in memory and then used to attempt a connection for command and control. By following the data we can identify the code blocks that touch and operate on it even if the data is in its n^{th} generation and morphed multiple times. Dataflow tracing follow and record many data mutations and data movements. Tying the data together gives a more complete picture.

III.D.7 Belief Reasoning and Inference Node (BRaIN)

So we have an input layer that consists of nodes that are the traits of software. The output layer would consist of nodes that represent what the software is, i.e. malware, spyware, virus, trojan, safe software, etc.

The DS Dempster-schaffer network would be able to show unknowns by having all of the input nodes having a high value for unknown. Viewing the internal structure of the belief network will reveal where the logic breaks down in trying to identify the unknown. For example, if the input layer shows that there is no significant traits that are discernible then this would indicate that there is a lack of information on this type of software. There could also be a mid level indicator that would show there is a lack of information on who created this software, which in turn would fail to identify this as safe software. Basically, the network itself is a tool in preforming analysis on the data. Another approach is to use data mining to correlate the unknowns to potentially knowns.

Research and develop an expert or AI model that can be trained and used to classify a malware object into categories. This will require processing a large set of known malware and a large set of known "clean" applications and code so that the model can reliably judge the intent of a given binary. A stochastic approach, such as a Belief inference model, can be matched with the probabilities learned and weights given to individual traits and behaviors.

Belief analysis is better thought of as probability theory. It is a model that can use the probability of events to calculate the probability of a more complex probability. The simplest examples are usually given as a deck of cards. The probability of drawing a spade from a normal deck of cards is 13 in 52 or 1 in 4. The probability of drawing a second spade is 12 in 51, or 4 in 17 times the probability of drawing the first, 1/4*4/17= 1/17 (0.0588235…). In Belief terms, the unconditional probability of the event (a card being a spade), with no additional knowledge or events, is 1 in 4. The conditional probability of an event (drawing a second spade), requires some additional evidence to compute (that we previously drew a spade). Belief probabilities are either computed analytically, or sampled empirically. Every possible event and potential evidence increases the complexity of Belief calculations, but is also likely to increase the accuracy and improve the understanding of the relationship between events and evidence. For our system, we will likely be using empirically sampled traits and behaviors and conditional probabilities between them to determine the probability of a binary being malicious or not malicious. [that was a very simplistic explanation of Belief reasoning, there is a lot more that could be explained, such as negative information, avoiding circular reasoning, joint probabilities, belief networks, etc]

Bayes' theorem shows the relation between one conditional probability and its inverse; for example, the probability of a hypothesis given observed evidence and the probability of that evidence given the hypothesis. The key idea is that the probability of event A given event B depends not only on the relationship between A and B but on the absolute probability of A independent of B, and the absolute probability of B independent of A.

Although Belief networks are often used to represent causal relationships, this need not be the case. A causal network is a Belief network with an explicit requirement that the relationships be causal. The additional semantics of the causal networks specify that if a node *X* is actively caused to be in a given state *x*, then the probability density function changes to the one of the network obtained by cutting the links from *X*'s parents to *X*, and setting *X* to the caused value *x*. Using these semantics, one can predict the impact of external interventions from data obtained prior to intervention.

Because a Belief network is a complete model for the variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to find out updated knowledge of the state of a subset of variables when other variables are observed. This process of computing the posterior sufficient statistic Bayes' theorem to complex problems. The posterior gives a universal for detection applications, when one wants to choose values for the variable subset, which minimize some expected loss function, for instance the probability of decision error. A Belief network can thus be considered a mechanism for automatically applying Bayes' theorem to complex problems.

The most common exact inference methods are: variable elimination, which eliminates the non-observed non-query variables one by one by distributing the sum over the product; clique tree propagation, which caches the computation so that many variables can be queried at one time and new evidence can be propagated quickly; and recursive conditioning, which allows for a space-time tradeoff and matches the efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in the network's treewidth.

The purpose of the Belief Reasoning Engine is to encode our prior knowledge about traits and genomes and to provide a mechanism to reason over that prior knowledge when new evidence is collected. The model construction process involves: identifying the evidence with discriminatory value, collecting that evidence, and constructing the model. Models for different malware will have some common elements and some unique elements. The goal for the model design is to maximize accuracy and generality. Generality is important so that each type of malware does not require a unique model, which would increase the effort to build the models and reduces the chances of detecting malware variants.

Dempster–Shafer theory is a generalization of the Bayesian theory of subjective probability; whereas the latter requires probabilities for each question of interest, Bayesian functions base degrees of belief for one question on the probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ depends on how closely the two questions are related. Put another way, it is a way of representing epistemic plausibility but it can yield answers which contradict those arrived at using probability theory.

Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule for combining such degrees of belief when they are based on independent items of evidence. In essence, the degree of belief in a proposition depends primarily upon the number of answers containing the proposition, and the subjective probability of each answer. Also contributing are the rules of combination that reflect general assumptions about the data.

In this formalism a degree of belief is represented as a belief function rather than a Belief probability distribution. Probability values are assigned to sets of possibilities rather than single events. Beliefs corresponding to independent pieces of information are combined using Dempster's rule of combination, which is a generalization of the special case of Bayes' theorem where events are independent. The probability masses from propositions that contradict each other can also be used to obtain a measure of how much conflict there is in a system. This measure has been used as a criterion for clustering multiple pieces of seemingly conflicting evidence around competing hypotheses. One of the computational advantages of the Dempster–Shafer framework is that priors and conditionals need not be specified, unlike Belief methods, which often use a symmetry argument to assign prior probabilities to random variables. However, any information contained in the missing priors and conditionals is not used in the Dempster–Shafer framework unless it can be obtained indirectly. Dempster–Shafer theory allows one to specify a degree of ignorance in this situation instead of being forced to supply prior probabilities, which add to unity.

III.E Comparison with Other Research

While there are many specific challenges related to automated malware analysis there are three main areas of research that are at the heart of this challenge:

- Trait based analysis of malware
- Increased execution of code paths
- Automated analysis of malware

The majority of trait based analysis capabilities, which are few, focus on providing textual information to the user on highlighted behaviors identified in an analyzed specimen. UCBerkley's Anubis Sunbelt Security's CWSandbox are probably the best examples of working capabilities in this area. In research there have been hypothesis made that suggest mathematical models for analyzing behaviors of malware, such as the MIST model presented in [Trinius, 2009] which describes a high level categorization of malware exhibited behaviors such as; thread, virtual memory, Winsock and some associated arguments. While this method could be successful at identifying gross functionality the model lacks a level of detail to be highly capable of determining to a level of detail malware function, behaviors, and intent. Our approach starts by developing a library of very detailed, mathematically calculable and human readable traits that describe discrete functions and behaviors of malware, not in the order of tens of traits but in the order of thousands of traits. The traits library combined with a patterns library to discern relationships between traits will give us a much higher fidelity capability. The challenge is the level of detail and understanding required to build the libraries is much more significant.

Increased execution of code paths has traditionally been accomplished through a combination of static binary analysis of branch points and brute force attempts using interactive debuggers. There is no existing technology that exercises branch points effectively. There does exist promising research in taint analysis [Yin, 2009], which involves instrumenting the system to monitor data flows of known variables as they flow through an executed binary.

Lastly, completely automated analysis of malware is something that has been research and for which many whitepapers are written with varying levels of specificity

indicating advantages and disadvantages of the proposed effort.

III.F Previous Accomplishments

The HBGary Federal Team brings significant experience and capabilities directly related to the objectives of the Cyber Genome Program with many successfully executed contracts in related areas for the Federal Government and Department of Defense (DoD). To demonstrate our ability to successfully execute a contract under DARPA's Cyber Genome Program we have selected one past performance citation from each of the team members.

III.F.1 HBGary Past Performance

Offeror Name: HBGary and HBGary Federal	Customer Organization: DHS Science and Technology Directorate	
Program Manager: Douglas Maughan	Address: 1120 Vermont Ave NW 8th Floor, Washington, DC 20528	
	Phone Number: 202-254-6145	
Contracting Officer: Doreen Vera-Cross	Address: P.O. Box 12924, Fort Huachuca, AZ 85670	
	Phone Number: 520-533-8993	
Contract Type: SBIR Phase II	Contract Value: $975,000	Dec 2007 – Nov 2010
Description of Worked Performed		
While most researchers approach the botnet problem by examining network traffic, HBGary chose host based examination because the bot (malware) must reside on the host in memory to execute. Our research focused on physical memory forensics including imaging memory, reconstructing memory and analyzing the recovered digital objects. Bayesian Reasoning Networks were explored to automate and scale the reasoning of security subject matter experts. Funding was added to research tools for automated Windows registry forensics and to provide training to law enforcement agencies to aid technology transition		
Relevance to DCG Technical Area 1		
The automated physical memory forensics and Bayesian Reasoning Networks modeling from this contract will be directly applicable to new research proposed for the Cyber Genome Program.		

III.F.2 Pikewerks Past Performance

Offeror Name: Pikewerks	Customer Organization: Air Force Research Laboratory	
Program Manager: Dr. David Kapp	Address: 2310 Eighth Street, Bldg 167, Wright-Patterson AFB, OH 45433	
	Phone Number: 937-320-9068 x130	
Contracting Officer: Erika Lindsey	Address: 2310 Eighth Street, Bldg 167, Wright-Patterson AFB, OH 45433	
	Phone Number: 937-255-3379	
Contract Type: CPFF	Contract Value: $750,000	PoP: Aug 2008 – Aug 2010

Description of Worked Performed

Anti-Forensics is the art and practice of obscuring data storage, transmission, and execution in such a way that it remains hidden from even a professional, dedicated examiner. Traditionally, hackers have used anti-forensic methods as a means of hiding their tools, techniques, and identities from forensic investigators. However, anti-forensic methodologies can also be adopted for defensive purposes. In particular, Anti-Forensic techniques have the ability to greatly increase the level of effort required to reverse-engineer malicious code. This is especially useful when the attacker has full access to the memory, disk, and possibly even the processor of a computer system running the protection software.

For this effort, Pikewerks has identified a number of anti-forensic research areas that would significantly enhance the confidentiality and integrity of executable code, data, and cryptographic materials through all stages of operation: at rest, in transit, and during execution. These areas include novel out-of-band storage and transmission techniques within Commercial Off The Shelf (COTS) computers, which go beyond the highest level of access available to an attacker and thus dramatically increase the level of effort required to fully identify, understand, or reverse-engineer the underlying code. The end goal of this development effort is a diverse suite of innovative anti-forensic capabilities that can be easily integrated into, and deployed with, technologies where stealth is critical.

Relevance to DCG Technical Area 1

This effort has resulted in the identification of anti-forensic capabilities that could be employed by sophisticated malware analysis authors, like the kind the Cyber GNOME Project is expected to engage. This effort is particularly useful to the DCG effort as it demonstrates the advanced research and development ongoing within Pikewerks Corporation. For the DCG effort revolutionary methods and techniques must be employed to analyze sophisticated malware that will in the future likely employ many of the techniques being studied by Pikewerks. Utilizing this research will assist in developing methods for identifying, analyzing, and relating sophisticated anti-forensic techniques within malware. The approaches developed include anti-forensic file system storage techniques, indirect function hooking, memory protection techniques using processor debug registers, and BIOS-based anti-forensic strategies. As part of the development of these techniques, Pikewerks has written several kernel modules and custom analysis capabilities for Windows and Linux that both characterize and detect sophisticated anti-forensic techniques.

III.F.3 GDAIS Past Performance

Offeror Name: GDAIS	Customer Organization: Defense Cyber Crime Center (DC3)	
Program Manager: Mike Buratowski	Address: 911 Elkridge Landing Road, Linthicum, MD 21090	
	Phone Number: 410-981-0117	
Contracting Officer: Jim Hayes	Address: 2100 Crystal Drive, Suite 300, Arlington, VA 22202	
	Phone Number: 703-605-3600	
Contract Type: T&M	Contract Value: $98M	PoP: Oct 2001 – Feb 2012

Description of Worked Performed

Department of Defense Cyber Crime Center (DC3) is a $126M multi-year T&M contract in support of the Air Force Office of Special Investigations (AFOSI). Since 2001, the GD Team has been the prime contractor for the Department of Defense Computer Forensics Laboratory (DCFL). In this capacity, the GD Team has conducted extensive network intrusion examinations and generated detailed reports documenting the intrusions. The DCFL, and DoD Cyber Crime Institute (DCCI) all fall under this contract.

Business Relationships & Customer Satisfaction: The GD management team provided the leadership that organized, planned, and managed the resources for the contract's major projects. Since careers and legal convictions are dependent upon our findings, we insist on the highest standards of quality and cross-check. The

GD Team is tightly integrated with the DC3 workforce of Government and Military personnel and work as equals in all facets of forensic support. The GD Team provides onsite program management at the DC3 for all contractor and subcontractor work. The Program Manager manages a staff of 140 personnel consisting of General Dynamics engineers, technicians, support personnel, and subcontractors. In March 2007, General Dynamics was awarded a new, 1-year (plus four option years) contract to provide Computer Forensic Examination support as well as Research, Development, Testing and Evaluation for computer forensic hardware and software.

Cost, Schedule & Timeliness: The GD Team has exceeded Government expectations by completing over 2,500 examinations, providing expert testimony in over 100 court proceedings (both CONUS and OCONUS), and serving as the DoD authority on electronic media forensics. DC3 Incident Response Support has experience with responses involving single system through large networks with enormous data storage capabilities. In its role, the GD Team has created a Virtual Analysis Environment where various system configurations including installed software packages and patch levels are already saved as Virtual Machines. The examiner can execute the known malicious logic within a system that is configured exactly how the compromised system would have been at the time of an intrusion.

Key Personnel: The GD Team accounts for over 80 percent of the personnel that perform data recovery, imaging and extraction, and forensic examinations in support of criminal, fraud, counterintelligence, data recovery, terrorism, and safety investigations in DC3. The team currently consists of 19 Cyber Intelligence Analysts, 13 Forensic Technicians, 48 Forensic Examiners, 15 Software Developers, and 5 Forensic Managers that perform casework for DC3.

Relevance to DCG Technical Area 1

This program has provided GDAIS with the operational knowledge and expertise of the latest intrusions and cyber threats seeing in DoD and Defense Industrial Base networks. In turn, it has provided GDAIS with the capabilities and knowledge to detect these cyber threats and their artifacts by using many of the forensics and reverse engineering capabilities within our analysis and R&D team. Since the number of intrusion cases has increase exponentially at DC3, we had the need to start performing automated behavior analysis and correlation between malware binaries. Within the DCFL/Intrusions Section, our engineers and computer scientist are developing a capability to automatically correlate these malicious binaries against malware found in previous intrusion cases. This is done with the use of IDA Pro and various fuzzy hashing techniques to disassemble the malicious binaries into individual function and perform correlation against the malware obtained through the many different intrusion cases. By using open source, freeware, and government sponsored tools they have also developed a capability to submit malicious binaries to perform automated behavioral analysis. This is the type of capabilities that together with our vast knowledge of the latest intrusions, GDAIS could leverage and enhanced for the DARPA Cyber Genome program. From the DCFL/NCIJTF perspective, our intelligence analysts use the analysis report generated by our DCFL\IA examiners to perform additional correlation against various events and data. Once this is done, reports and signatures (intrusion indicators) are distributed to the community. The DCCI R&D team is constantly collaborating with different DoD, academia, and industry organization to learn about their effort and share tools for addition into our DC3 operations. Many of these tools are tested and validated by our DCCI T&E team to verify that the results are accurate and reliable.

For technical area one of the DARPA Cyber Genome program, GDAIS, together with their partners, will employ revolutionary techniques to exploits our collective knowledge and expertise to automatically ingest these malicious binaries and provide correlation, lineage, and provenance in order to gain a better understanding of software evolution, detect zero-day malware, and when possible determine attribution.

III.F.4 SRI International

Offeror Name: SRI International	Customer Organization: Army Research Office
Program Manager:	Address: 4300 S. Miami Blvd, Durham, NC 27703

Cliff Wang	Phone Number: 919-549-4207	
Contracting Officer: Kathy Terry	Address: P.O. Box 12211, Research Triangle, NC 27709	
	Phone Number: 919-549-4337	
Contract Type: Grant	Contract Value: $13.4M	PoP: Jun 2006 – Jul 2010

Description of Worked Performed

Phillip Porras is the Principal Investigator of the Army Research Office sponsored Cyber-TA Project. Cyber-TA is an ongoing 5-year research project to develop the next-generation of real-time national-scale Internet-threat analysis technologies. Our team has developed many new sophisticated antimalware and malware tracking technologies, produced over 50 publications in scientific peer reviewed venues, and has deployed its technologies widely across DoD and the U.S. Government. The Cyber-TA research project has brought together many of the world's most established researchers across the fields of data privacy, cryptography, malware and intrusion detection research, as well as operational experts in Internet-scale sensor management, to develop leading edge solutions to the evolving threat of increasingly virulent and wide-spread self-propagating malicious software. Examples of Cyber-TA research technologies include:

- Eureka – A binary unpacking and decompilation system designed to overcome a broad spectrum of malware binary logic protection services: http://eureka.cyber-ta.org
- BLADE – A system to immunize Windows platforms from malicious drive-by malware exploits: http://www.blade-defender.org
- Highly Predictive Blacklists – A link-analysis-based IP blacklist production system for producing high-quality network blacklists: http://www.cyber-ta.org/releases/HPB/
- BotHunter – A network-based host infection diagnosis system: http://www.bothunter.net/
- Malware Threat Center – A portal for tracking Internet malware threats across the Internet: http://mtc.sri.com
- Malware Cluster Lab – An example of SRI's experience in appling malware forensic clustering to detect malware binary lineage is available at http://cgi.mtc.sri.com/Cluster-Lab/, and an example of our ability to conduct a quantifiable comparison of pair-wise binary logic within two malware binary samples that employ multi-layered packing is available at http://mtc.sri.com/Conficker/addendumC/HMA_Compare_ConfB2_ConfC/.

A Cyber-TA project overview description is available at: http://www.cyber-ta.org/pubs/IEEE-SnP-Magazine-CTA_Nov2006.pdf

Relevance to DCG Technical Area 1

Cyber-TA has provided an ongoing resource for SRI's Computer Science Laboratory to conduct both breadth and depth research in understanding and combating the modern Internet crimeware epidemic. Of particular relevance to DCG is the extensive Cyber-TA research that our team has produced in the area of binary unpacking, disassembly, decompilation, and deobfuscation. We have demonstrated our advanced deobfuscation techniques in work such as (http://mtc.sri.com/Conficker/P2P/index.html), which is to our knowledge the only published description of the multi-layered obfuscated code base of the Conficker P2P subsystem. An example of our ability to handle mobile malware binary reverse engineering on non-x86 binaries is available at http://mtc.sri.com/iPhone/.

III.F.5 AVI/Secure Decisions

Offeror Name: AVI-Secure Decisions	Customer Organization: AFRL / IARPA / NSA
Program Manager:	Address: 525 Brooks Road, Rome, NY 13441

Walter Tirenin	Phone Number: 315-330-1871	
Contracting Officer: Rebecca Willsey	Address: 26 Electronics Parkway, Rome, NY 13441	
	Phone Number: 315-330-4710	
Contract Type: BAA	Contract Value: $2.3M	PoP: Sep 2005 – Dec 2008

Description of Worked Performed

VIAassist is a visualization framework used by computer security specialists to ensure the security of computer networks. It was developed to visualize NetFlow data, and is currently used for classified applications by the IC and being modified for adoption by DHS in US-CERT. In addition to NetFlow data, VIAssist can visualize intrusion detection and other data sources. VIAssist converts network data into a collection of graphical representations to make it easier to see patterns and trends. This technique takes advantage of the innate ability of humans to perceive patterns in pictures that they might otherwise miss when looking at raw data. It provides IC analysts and cyberdefense personnel with the following capabilities that have enhanced the overall mission, meeting the performance, cost and schedule criteria.

- **Provide workflow continuity & collaboration.** Analysts record observations, and shared annotations allow users to collaborate with colleagues about their findings.
- **Provide effective reporting.** Through the use of the Report Designer and pre-defined report templates, VIAssist streamlines report building for analysts.
- **Provide global & detailed situational awareness.** Dual monitor displays provide a global, summarized view of trends, as well as a focused view of specific incidents.
- **Provide multiple views of the same data.** Multiple coordinated views of the data are provided to make it easier to identify anomalies, relationships and interdependencies between data points.

- **Correlate multiple data sources.** Using an intermediary data store, integrates with and visualizes multiple disparate data sources, such as firewall logs, IDS data and NetFlow data.
- **Aggregate data.** Through the use of Smart Aggregation technology, effectively displays voluminous data by visually aggregating data into meaningful visualizations with drill-down capability and in so doing, reduce load on system and response time. .
- **Filter data.** Through the use of an advanced Expression Builder, filters data based upon various pre-defined or complex user-defined criteria, allowing analysts to focus on specific data, to the exclusion of the mass of “noise” that can often obscure security risks.

Relevance to DCG Technical Area 1

Specific technologies developed for VIAssist that support smart data aggregation may be leveraged to assist in providing compelling and scalable visualizations to support malware analysis.

III.G Place of Performance, Facilities, and Locations

The HBGary Federal team will perform work at their individual office locations. We propose no classified work, but will be able to support classified discussions, meetings and briefings at government facilities. Each team member has a primary location and may have a secondary location in which they will perform research and development. A summary listing is provided in Table #.

Company	Location
HBGary Federal	Sacramento, CA
HBGary	Sacramento, CA
Pikewerks	Alexandria, VA
SRI International	Menlo Park, CA
Secure Decisions	Northport, NY
General Dynamics	Centennial, Co

Table #. Description of Facilities

III.H Detailed Support (Including Teaming Agreements)

HBGary Federal has fully executed teaming agreements with following companies for the purposes of preparing a written proposal for DARPA-BAA-10-36_Cyber_Genome and for the execution of said contract upon award (copies of teaming agreements available upon request): HBGary, Inc.; Pikewerks; General Dynamics AIS; SRI International; and AVI/SecureDecisions.

III.I Cost, Schedules and Measurable Milestones

including estimates of cost for each task in each year of the effort delineated by the primes and major subcontractors, total cost, and any company cost share. **Note: Measurable milestones should capture key development points in tasks and should be clearly articulated and defined in time relative to start of effort.** These milestones should enable and support a decision for the next part of the effort. <u>Additional interim non-critical management milestones are also highly encouraged at regular intervals.</u>

Where the effort consists of multiple portions that could reasonably be partitioned for purposes of funding, these should be identified as options with separate cost estimates for each. Additionally, proposals should clearly explain the technical approach(es) that will be employed to meet or exceed each program metric and provide ample justification as to why the approach(es) is/are feasible. **Note: Task descriptions related to the technical approach and associated technical elements need to be complete and clearly related to satisfying the program metrics as stated in Section 1.2.1.**

Task	Contractor	Year	Cost	Success Criteria
Task1	SRI	1	$499,997	
	Pikewerks		$326,083	
	HBGary Federal		$0	
			$0	
	SRI	2	499,925	
	Pikewerks		229,100	
	HBGary Federal		$0	
			$0	
	SRI	3	$543,018	
	Pikewerks		$119,227	
	HBGary Federal		$0	
			$0	
	SRI	4	$557,007	
	Pikewerks		$89505	
	HBGary Federal		$0	
			$0	
	Total Task 1		**$0**	

Task 2	HBGary Federal	1		
	HBGary Federal	2		
	HBGary Federal	3		
	HBGary Federal	4		
	Total Task 2		**$0**	
Task 3	Secure Decisions	1	$435,937	
	GDAIS		$26,119	
			$462056	
	Secure Decisions	2	$465,727	
	GDAIS		$26789	
			492,516	
	Total Task 3		**$954,572**	
Task 4	HBGary Federal	2		
	HBGary			
	Pikewerks		$52,346	
			$0	
	HBGary Federal	3		
	HBGary			
	Pikewerks		$119,227	
			$0	
	HBGary Federal	4		
	HBGary		$0	
	Pikewerks			
			$0	
	Total Task 4		**$0**	
Task 5	HBGary Federal	1		
	HBGary			
	Pikewerks		$118,369	
	General Dynamics		$80,366	
			$0	
	HBGary Federal	2		
	HBGary			
	Pikewerks		$52,346	
	General Dynamics		$82,428	
			$0	
	HBGary Fedreal	3		
	HBGary			
	Pikewerks		$119.227	
	General Dynamics		$84,795	
			$0	
	HBGary Federal	4		
	HBGary			
	Pikewerks		$122,804	
	General Dynamics		$87,235	
			$0	
	Total Task 5		**$0**	
Task 6	HBGary	2		
	Pikewerks		$129,224	
			$0	
	HBGary	3		

	Pikewerks		$119,227	
			$0	
	HBGary	4		
	Pikewerks		$122,804	
			$0	
			$0	
Task 7	HBGary Federal	3		
	HBGary Federal	4		
			$0	

III.J Data Description

HBGary Federal subscribes to commercial malware feeds and has an existing 500GB unique sample malware repository that will be used for this effort. We will also acquire new feeds and develop malware harvesters to find and capture new malware that is not available in the feeds. Collection of new malware will be through seemingly normal web-based activities. The malware objects are binaries, PDF, documents that are or contain malware. We will ensure the feeds we subscribe to acquire malware through legal, non-intrusive means.

Section IV. Additional Information

A brief bibliography of relevant technical papers and research notes (published and unpublished) that document the technical ideas upon which the proposal is based. Copies of not more than three (3) relevant papers can be included in the submission.