SANDIA REPORT

SAND2014-0446
Unlimited Release
Printed January 2014

Hybrid Methods for Cybersecurity Analysis LDRD Final Report

Warren L. Davis IV and Daniel M. Dunlavy

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-0446
Unlimited Release
Printed January 2014

Hybrid Methods for Cybersecurity Analysis LDRD Final Report

Warren L. Davis IV
Scalable Analysis & Visualization
Sandia National Laboratories, Albuquerque, NM 87185-1327
Email: wldavis@sandia.gov

Daniel M. Dunlavy
Analytics Department
Sandia National Laboratories, Albuquerque, NM 87185-1327
Email: dmdunla@sandia.gov

Abstract

Early 2010 saw a significant change in adversarial techniques aimed at network intrusion: a shift from malware delivered via email attachments toward the use of hidden, embedded hyperlinks to initiate sequences of downloads and interactions with web sites and network servers containing malicious software. Enterprise security groups were well poised and experienced in defending the former attacks, but the new types of attacks were larger in number, more challenging to detect, dynamic in nature, and required the development of new technologies and analytic capabilities.

The Hybrid LDRD project was aimed at delivering new capabilities in large-scale data modeling and analysis to enterprise security operators and analysts and understanding the challenges of detection and prevention of emerging cybersecurity threats. Leveraging previous LDRD research efforts and capabilities in large-scale relational data analysis, large-scale discrete data analysis and visualization, and streaming data analysis, new modeling and analysis capabilities were quickly brought to bear on the problems in email phishing and spear phishing attacks in the Sandia enterprise security operational groups at the onset of the Hybrid project. As part of this project, a software development and deployment framework was created within the security analyst workflow tool sets to facilitate the delivery and testing of new capabilities as they became available, and machine learning algorithms were developed to address the challenge of dynamic threats. Furthermore, researchers from the Hybrid project were embedded in the security analyst groups for almost a full year, engaged in daily operational activities and routines, creating an atmosphere of trust and collaboration between the researchers and security personnel.

The Hybrid project has altered the way that research ideas can be incorporated into the production environments of Sandias enterprise security groups, reducing time to deployment from months and years to hours and days for the application of new modeling and analysis capabilities to emerging threats. The development and deployment framework has been generalized into the Hybrid Framework and incorporated into several LDRD, WFO, and DOE/CSL projects and proposals. And most importantly, the Hybrid project has provided Sandia security analysts with new, scalable, extensible analytic capabilities that have resulted in alerts not detectable using their previous workflow tool sets.

Acknowledgments

This work was fully supported by Sandia's Laboratory Directed Research & Development (LDRD) program.

Contents

This page intentionally left blank.

1 Introduction

Early 2010 saw a significant change in adversarial techniques aimed at network intrusion: a shift from malware delivered via email attachments toward the use of hidden, embedded hyperlinks to initiate sequences of downloads and interactions with web sites and network servers containing malicious software. Enterprise security groups were well poised and experienced in defending the former attacks, but the new types of attacks were larger in number, more challenging to detect, dynamic in nature, and required the development of new technologies and analytic capabilities.

The Hybrid LDRD project was aimed at delivering new capabilities in large-scale data modeling and analysis to enterprise security operators and analysts and understanding the challenges of detection and prevention of emerging cybersecurity threats. Leveraging previous LDRD research efforts and capabilities in large-scale relational data analysis (Network Discovery, Characterization, and Prediction Grand Challenge LDRD project, FY08-10), large-scale discrete data analysis and visualization (Scalable Solutions for Processing and Searching Very Large Document Collections, FY08-10), and streaming data analysis (Streaming Data Analysis for Cybersecurity, FY10-12), new modeling and analysis capabilities were quickly brought to bear on the problems in email phishing and spear phishing attacks in the Sandia enterprise security operational groups at the onset of the Hybrid project. As part of this project, a software development and deployment framework was created within the security analyst workflow tool sets to facilitate the delivery and testing of new capabilities as they became available, and machine learning algorithms were developed to address the challenge of dynamic threats. Furthermore, researchers from the Hybrid project were embedded in the security analyst groups for almost a full year, engaged in daily operational activities and routines, creating an atmosphere of trust and collaboration between the researchers and security personnel.

The Hybrid project has altered the way that research ideas can be incorporated into the production environments of Sandias enterprise security groups, reducing time to deployment from months and years to hours and days for the application of new modeling and analysis capabilities to emerging threats. The development and deployment framework has been generalized into the Hybrid Framework and incorporated into several LDRD, WFO, and DOE/CSL projects and proposals. And most importantly, the Hybrid project has provided Sandia security analysts with new, scalable, extensible analytic capabilities that have resulted in alerts not detectable using their previous workflow tool sets.

1.1 Motivation

In previous years, the main threats from emails were in the form of attachments. Embedded executables, zip files, pdfs, etc. contained malicious code. This malware, when executed by the recipient of the email, would attempt to breach our cyber defenses. Sandia's cybersecurity personnel and appliances became rather skilled at detecting such emails and filtering out these threats. However, attackers adapted and chose a new methodology.

Instead of attaching including the malicious payload in the email, the attackers would instead send links to external sites which would host the malicious code. The links would be embedded in emails that were crafted to look benign and attractive. Simply by following the link, the recipient exposes himself to a specified danger. This technique is called *phishing*, and has become one of the foremost types of attacks facing Sandia's cyber defenders.

In recent years, Sandia has seen an increase of phishing attacks. Classic examples of this are links that attempt to mimic a reputable site, such as a financial institution, and thereby obtain sensitive information about the recipient. However, Sandia is more concerned about web accessible links which lead to some destructive mobile code that, when accessed, allows adversaries to attack our networks. These attacks are more difficult to detect, as there is no attachment which can be scanned with an antivirus attachment. Finding these types of attacks in our emails is a much more tedious task, and drastically increases the difficulty of the work for Sandia's cyber analysts. In addition, as there are many emails that come in with

embedded links, this also increases the number of events that require analyst investigation

1.2 Related Work

The challenges associated with the processing and analyses of a live network stream are formidable. There are a myriad of open source toolkits used for the ingestion and display of network packet data; Snort [1], Wireshark (http://www.wireshark.org), and Bro IDS [2] are a few of the popular ones, with more listed at http://sectools.org. Such tools often provide a comprehensive set of filters or rules that can be applied to the network stream, which in the case of Snort can include upwards of 20,000 rules. Work focused on the analysis of network logs is also common: Splunk [3] is a popular commercial tool and Analysis Farm [4] uses a NoSQL database to improve aggregation and query performance. Although powerful, these low-level localized rule-based approaches often break down for higher-level analysis functions such as trend analysis of organizational email traffic or identification of high-risk web behavior.

The other extreme to these constrained streaming tools is to save the network captures to disk and utilize one of the analytically rich, publicly available toolkits such as WEKA [5], Orange [6], Titan Toolkit [7], etc., on the historical data. The significant downside to this file-based approach is that the analysis becomes forensic in nature and the developers may find themselves describing their new algorithm results against static data sources perhaps limiting the efficacy of the results when compared to approaches running on live production network streams.

The NoSQL database community is actively using these data stores in increasingly novel ways [8, 9, 10, 11]; in particular, the work for this LDRD project was focused on the use of NoSQL databases to cross the chasm between traditional offline analysis and real-time network systems. This approach enables the application of analytical models to streaming network data with the results presented to a network defender within 10-20 seconds. In addition, our approach provides a flexible environment where different programming languages and scripts are welcome, components are interchangeable, and most importantly its remarkably resilient to unstable research-focused code.

1.3 Solution Overview

To address the problem of phishing email detection, we have developed a framework for processing network packet data, storing and managing data, data models and analysis models, training analysis models, and evaluating analysis models.

Figure 1 presents an illustration of this framework as it is applied to email data. On the top left, network packets are collected and SMTP sessions are reconstituted using one of several capabilities; we have tested the Bro IDS system [2] and another proprietary Sandia capability. Results (email messages and associated packet metadata) are then stored in a NoSQL database (top center). Email messages are then processed for features and optional used in training or fitting data or analysis models (top right). We have explored several types of data models (Latent Dirichlet Allocation [12] and Latent Semantic Analysis [13]) and analysis models (k-means clustering, k-medoids clustering, k-nearest neighbor classification, and multilayer perceptron classification). Once the models have been trained or fit, they are serialized and place in the NoSQL database alongside the data. These models are evaluated in real-time on incoming email and the results are output to an analyst's tool (e.g., Splunk as shown in Figure 1).

One of the approaches for analysis that we have explored in this project incorporates analyst feedback into model training. For example, we have developed Splunk apps that present the results of model evaluations on email (e.g., a label of phish or not-phish) and allow analysts to correct predicted labels. These correctly annotated labels are then used to train new versions of the models, so that analysts' expertise associated with the newly labels email message is reflected in the label prediction models.

We have also explored other Splunk app prototypes for general network packet analysis and visualization

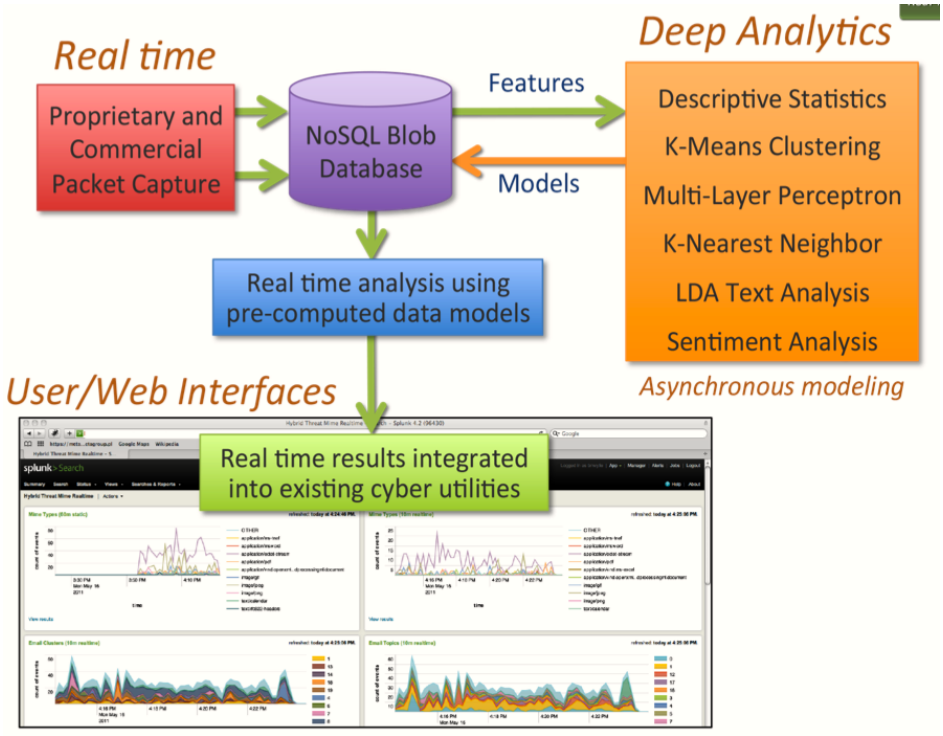

Figure 1: Overview of the framework and analytic approaches developed under the Hybrid Methods for Cybersecurity Analysis LDRD Project.

(i.e., beyond email messages) and several other data analysis applications (see Section 3.2 for more details).

Initial Framework Design

The initial solution attempt included a few major components for processing raw data, extracting features from the data, and training/evaluating predictive models. Each of these components was responsible for managing its own data, interactions with a data store, and all functionality associated with a particular task (i.e., processing, feature ext ration, modeling). Although this provided the means for a working system for addressing the problem of labeling new phishing email given examples of previously encountered phishing email, there were several drawbacks to the approach.

The use of the NoSQL database called CouchDB was initially used as a data store for our framework. All components were responsible for managing connections to CouchDB and storing and retrieving data. With few components, this was an acceptable solution. However, as we turned to parallel email processing and feature extraction through the use of the `multiprocess` package, the lack of locking in CouchDB presented new challenges in processing large amounts of data.

The initial system was also not extensible as designed, as all functionality of a particular type was run on every data instance. For example, all features were extracted for all email messages, no matter what type of analysis was subsequently performed. Furthermore, all types of analyses were performed for all email messages. For example, email features were modeled using Latent Dirichlet Allocation [12], clustered using k-medoids, and classified as a known type of email messages or as an "unknown" message type. Processing all data for all types of analyses that might be required was unnecessary and led to a more modular, agile framework design to allow for multiple types of analyses simultaneously.

Although there were problems with the initial design, the framework was capable of processing, extracting features, and predicting categories of several hundred thousand email messages per day.

Modular Framework Design

About midway through the project, a complete refactoring of the framework was performed. This refactor included design of classes for the data store (`db`), data objects (`data_blob`), data encoding logging (`encoding`), data and analysis models (`model`), multiprocessing (`mp_pool`), pipeline components (`worker`), and pipeline management (`manager`), and logging (`logger`). The modular design supported more rapid prototyping of new analysis ideas, the ability to run multiple analyses simultaneously, and the ability to interface with capabilities external to the framework.

The components of the final modular system are described in the next section.

2 The Hybrid Framework

In this section, we describe the Hybrid Framework, the modular system for processing, modeling, and evaluating data.

2.1 NoSQL Data Store

One of the major components of the Hybrid Framework is the data store, where both data, models, and analysis artifacts are stored. Each module interacts with the data store to both request input and store output. This output can consist of data input and output (e.g., email messages and associated metadata and features), model parameters and information (e.g., type of model, how a model was constructed or trained, and specific parameters for a model instance), module artifacts (e.g., benchmarking information), or any combination of these.

The original data store in the Hybrid Framework was CouchDB [14], a NoSQL database. An important benefit in using a NoSQL database is that no schema is required to specify data to be stored. Such a flexibility is important in data analysis applications where data fields change over time. For example, in email applications, the fields appearing in email messages may include arbitrary tags (e.g., `X-Foo-Header`). With no schema requirement, the data store can store this email header field as a data instance field without any changes to the data store (beyond the specific data instance). Another benefit is that binary large objects (BLOBs) can be stored in CouchDB, which provides support for storing the tips of data required in our applications: data, models, and artifacts.

Due to some limitations with CouchDB (e.g., lack of support for generic run-time queries) and the use of MongoDB [15] in several of the main application groups working with the LDRD team led to a refactor of the data store interface. Specifically, a new `db` class was created as an interface to any back-end data store that satisfies the Hybrid Framework requirements of schemaless data storage, data locking, BLOB support, and efficient data parallelism. At the same time, a new data instance class, `data_blob`, was introduced to allow for more efficient storage and retrieval for large data instances. Specifically, `data_blob` instances contain both metadata (e.g., header fields) and data (e.g., am email message and its attachments).

2.2 Stateful Data Model

One of the challenges in processing and analyzing streaming data is keeping track of the state of any individual data instance (e.g., email message and associated metadata and features). In our framework, we opted to store stateful information as part of each data instance. Although this requires more storage, the benefits of knowing the state of a particular data instance in multiple-step, complicated analysis pipelines outweighed the storage and communication costs. For example, for modular designs where (references of) data instances are passed between modules, each module only needs to check the state of a data instance that is required for its own processing, reducing the complexity of individual modules. Furthermore, instead of executing serial pipelines on each data instance, the modules can pull data from the data store when its state indicates that it is ready to move to the next part of the processing or analysis pipeline, reducing the amount of time modules sit idle.

State in the Hybrid Framework is stored using boolean flags, where the field is the flag name and the value is `True` or `False`. In most of our applications to date, we primarily store module dependencies, and thus the presence of a state field is the indication of the state. For example, if feature extraction depends on packet sessionization, the module handling the packet sessionization could add a field called `packet_sessionized` to each data instance when the sessionization has been completed. The feature extraction module then only needs to request from the data store a new batch of data whose state include `packet_sessionized=True` to perform its tasks.

The Hybrid Framework supports more complex state information than simple boolean flags, but such state checking is handled through more complicated data store queries that would be specific to that particular state information. For example, counters could be used to store state for each data instance, but a module depending on that state would need to specify a query to the data store that included logic beyond just the presence of a particular field (e.g., `counter > 5`).

One other important benefit in using a stageful data model is that it provides support for data parallelism. This benefit has been leveraged via the `mp_pool` class in the Hybrid Framework, which supplies multithreading support for modules that can take advantage of data parallelism (i.e., can process data instances independently). Such parallelism allows for improved scalability in many of our applications to date without requiring users or module developers to develop their own multithreaded solutions.

2.3 Executor-Manager-Worker Model

Hybrid employs an Executor-Manager-Worker model for organization of functionality. These partition development in a way so as to encapsulate responsibilities, while providing standardization to facilitate generalization.

Workers

Workers are the most fundamental unit of processing in the Hybrid Framework. It is within the worker that documents are processed, either individually or en masse. The API of Hybrid allows a developer to write a specific worker without knowing anything about the underlying database. The retrieval and storage-side updating of the database document are handled elsewhere. In addition, the Hybrid API provides functionality to facilitate the retrieval of metadata from within the document in formats usable through standard pythonic conventions and toolkits such as SciKits and Titan.

Information produced within the worker can be added back into the documents. An example would be a document classification produced by projecting the document machine learning worker. This new information can be added either directly or in nested data structures that contain relevant metadata about the worker, such as the processing datetime stamp and version information.

For analysis algorithm developers and many researchers, the worker will be the primary area of development. Workers inherit a high degree of generic functionality, so for simple workers only one method will need to be written. More complex workers can be written by providing additional support functions, such as the generation of models.

Manager

Managers take, upon construction, a list of workers to employ. The manager schedules the workers themselves, and is responsible for balancing the task loads to each worker.

During processing, the manager spawns a number of threads as defined by the worker_threads parameter. In this way, data can be processed in parallel. As this multi-threading is handled by the manager, it absolves the Worker developer from the responsibility of thinking about this aspect. Of course, other types of parallelism can be achieved by building to those specific paradigms.

The managers receive information detailing what documents to pull for processing. These specifications are usually given in the form of a query specification– individual terms for a database such as MongoDB, or a view name for a database such as CouchDB. In addition, the manager is given a tag specification to give the documents after processing. Depending on the outcome of the workers, the tag will be given a value of complete or incomplete. An incomplete document might result from a version incompatibility or some

unexpected data format. With this differentiation, a manager knows not to try to reprocess problematic documents, but downstream processes know to leave the incomplete document alone. Incomplete documents can be handled manually by operators of the system, or by additional workers made specifically for automated resolution.

The manager class itself does not need to be modified by developers of a Hybrid pipeline. Instead, individual manager instances are created in the Executor, discussed below.

Executors

Executors are application side python programs where the individual Workers and Managers are instantiated along with the appropriate parameters. These programs typically contain command-line parsers, program file I/O, and other application-specific logic. In an executor, the developer instantiates workers using specific parameters. The developer then passes the workers to a manager instantiation, along with any manager parameters, such as the number of worker threads to use.

The development of Executors falls to those who are responsible for the flow of tasks and the overall design of the particular Hybrid pipeline. These will usually be domain experts, who are working with researchers to provide a solution to a specific problem.

In our use cases, we found it useful to provide configuration files to store the executor parameters. The specifics of these can then be shared with analysts and discussed in a more tangible format.

2.4 Modeling and Analysis

Using the Hybrid Framework API, various modeling and analysis algorithms were created. A few examples are as follows:

- Feature extraction: metadata, email processing, text processing, web data processing
- Descriptive statistics: attribute-based precision, recall, and F-measure
- Unsupervised ML: Latent Dirichlet Allocation (LDA), partitional and hierarchical clustering
- Supervised: k-Nearest Neighbor, Multi-Layer Perceptrons

These analysis algorithms support the creation and use of models, either offline or in near-real-time. They have been quite instrumental in increasing the situational-awareness of Sandia's cyber analysts. However, they were created to be general purpose algorithms, applicable to many types of data. The algorithms can be customized inline to fit the analysis characteristics of any specific problem (e.g., feature sets, model parameters and hyper-parameters, output artifacts, etc.). Currently, the primary algorithms are derived either from independent python code or from Sandia's Titan Toolkit. Work continues to integrate other analysis packages, such as SciKits and WEKA.

2.5 Application Integration

Bro IDS

To provide a data ingestion capability that was independent of Sandia's proprietary packet capture capabilities, we developed an interface in the Hybrid Framework to leverage the Bro IDS system [2] for packet capture and sessionization. This included a change to the Bro souce code to support direct interaction with CouchDB. This source code change was submitted to the Bro development team and included in releases as of version 1.7. Beyond the source code update, this engagement has led to new potential collaborations with Lawrence Berkeley National Laboratory (main group of users and developers of Bro within the DOE) and

the International Computer Science Institute (current set of developers supported by the NSF).

Splunk

One primary key to the success of the Hybrid Methods LDRD was the integration with Splunk, software used by Sandia's cyber analysts to search and index data from streaming sources. As our cyber experts were already familiar with this interface, it made sense to make our pipeline compatible with the Splunk utility. The Hybrid Toolkit contains a module specifically for integrating the results of our processing into a Splunk-accessible format, allowing the analysts to query the Hybrid Toolkit artifacts in the same way they would query other data.

In addition, we worked with Kitware, and external contractor, to develop visualizations and interfaces that could be embedded into Splunk. Visualizations included spatial graphs of incoming data, allowing analysts to visualize clusters of related information. We also created generic interfaces for No-SQL databases, allowing the analysts to explore and modify streaming documents from with Splunk itself.

3 Impact

The Hybrid Methods LDRD and the resultant Toolkit has had a significant impact on research and operations at Sandia. A few salient examples are discussed below:

3.1 Cybersecurity

The primary application of the Hybrid Toolkit has been in cybersecurity, addressing the problems endemic to the detection of malicious emails. To this end, the Hybrid Methods LDRD has been very successful. Some of the advancements that have been made are as follows:

Support of integrated offline and near-real-time analysis

The modeling and analysis models of the Hybrid Toolkit allow for analysis models to be developed offline without impeding the flow of cyber operations. The resultant models can then be used to process new data in near-real time. Analysts can present new data to the analysis modules, resulting in either immediate or scheduled recalculation of models. These models can be archived, shared, and compared very easily, facilitating information-sharing amongst cyber analysts.

Data loss prevention

Previous methods of analysis on email data were impeded because of processing constraints. Analysis that was too complex could cause data loss, which could be very detrimental from a situational-awareness perspective. The Hybrid Framework allows complex analysis without the concern of data loss, allowing cyber analysts to have a more complete picture of their cyber environment.

Rapid prototyping

One of the major advancements of the Hybrid Methods LDRD is that the resultant toolkit allows new prototypes to be deployed on real-time cyber data in hours, rather than the days it would have taken before hand. Algorithm researchers can focus on the creation of analysis models, ignoring many of the typical Extract-Transform-Load (ETL) and deployment challenges that they would face in the past. The flexible interface has, in fact, enabled cyber analysts to write their own analysis algorithms, exchanging data with the models that algorithm researchers are providing.

3.2 Other Applications

Grandmaster, P.I. : Nathan Fabian

Project Grandmaster is research designed to aid instruction and learning by analyzing a person's social media data. The project facilitates visual exploration, summarization, and understanding of large amounts of textual data. The Hybrid Framework was instrumental as a model for data retrieval and text analytics processing, and analysis of the requirements and constraints of Project Grandmaster gave valuable information that helped shape the Hybrid Framework itself. It is likely that the Hybrid Toolkit will be a part of the future development of Project Grandmaster.

Prevention of OUO Exfiltration, P.I. : JT McClain

The Hybrid Toolkit is being used in the development of an Automated Text Categorization Tool for scanning text within documents, including email, to identify sensitive information. The categorization tool could potentially be utilized by all government personnel to scan text they create to identify its sensitivity and to help prevent unauthorized releases. The application could be used stand-alone by personnel, or could be integrated as a part of an automated system for applications such as filtering outgoing emails.

Global Repository and Unified Natural Text Search (GRUNTS), P.I. : JT McClain

The Global Repository and Unified Natural Text Search (GRUNTS) utility is a distributed intelligent information retrieval system. It has been used in various research and operation capacities, and is particularly useful for cyber analysts. Utilizing components such as Apache Lucene and No-SQL databases, it allows flexible and scalable search functionality for its users. Although GRUNTS, itself, does not depend on the Hybrid Toolkit, it has been successfully used with data produced from Hybrid to provide enhanced information access to analysts. In addition, the Hybrid Toolkit provided valuable information to the GRUNTS project that helped shape the development of new functionality.

3.3 Follow-on Work

The Hybrid Methods LDRD continues to impact research at Sandia, inspiring new projects and follow-on research. A few of the more salient examples are detailed below:

Continued Cybersecurity Research

Because of the close joint participation between 9300 and 1400, the Hybrid Framework has become an integral part of various cyber-security activities. It has led to the facilitation of cyber operations and general increased situational awareness. Our cyber analysts are exploring additional ways that the framework can be useful. To this end, 9300 and 1400 are continuing to work closely to add additional functionality both to the Hybrid Toolkit and the the cyber assets that utilize Hybrid. Funding has been allocated to increase the efficiency and stability of the Hybrid Toolkit components and to research how machine learning algorithms can better be used to enhance cyber security.

DOE Cyber Sciences Laboratory (CSL)

Throughout 2012-2013, several meetings throughout the DOE were held to discuss the possible establishment of a Cyber Sciences Laboratory. Staff members from this project and others, along with several managers participated in these meeting, and were subsequently asked to lead sections of the proposal for work associated with the CSL. Danny Dunlavy led the proposal team for the Tools and Analytical Frameworks thrust under the Big Data & Behavioral Cyber Analytics program. This team consisted of staff members from LLNL, LANL, and PNNL, and the requested 1-year budget for the proposed work was $1,350K, with $900K specified for SNL. The proposed work included the development of a pilot program for deployment throughout the DOE complex of the Hybrid Framework and associated analytics developed in this project. Funding for the CSL and specifically the work associated with this project has yet to be determined as of the writing of this report.

Projects using NoSQL Data Stores

Although several projects have been using the Hybrid Framework directly, this LDRD project has also indirectly impacted several projects using NoSQL data stores as well. Specifically, Joe Ingram from this LDRD project, and the developer of the `db` interface class in the Hybrid Framework, has developed similar data stores for use in other research projects. Joe developed the data store interfaces that are used in the Host Watch LDRD (Joe Ingram, PI) and the Altert Triage LDRD projects (Justin Doak, PI). The development of all three data stores was guided by many of the successes (and failures) encountered with using NoSQL data stores for scalable data analysis in this LDRD project.

3.4 Publications and Presentations

Publications

- **Using NoSQL databases for streaming network analysis**, LDAV 2012 [16]
  Abstract: The high-volume, low-latency world of network traffic presents significant obstacles for complex analysis techniques. The unique challenge of adapting powerful but high-latency models to realtime network streams is the basis of our cyber security project. In this paper we discuss our use of NoSQL databases in a framework that enables the application of computationally expensive models against a real-time network data stream. We describe how this approach transforms the highly constrained (and sometimes arcane) world of real-time network analysis into a more developer friendly model that relaxes many of the traditional constraints associated with streaming data. Our primary use of the system is for conducting streaming text analysis and classification activities on a network link receiving 200,000 emails per day.

Presentations

- **Practicing Open Source**, 2011 O'Reilly Open Source Conference (OSCON) [17]
  Abstract: The pace of research and development is accelerating. Older methods of moving science from the laboratory, typically based on publication, technology transition and commercialization, are in many cases obstacles to rapid innovation. New approaches based on open practices can make dramatic impacts on the pace of innovation. In particular, the computational sciences benefit significantly from the use of open source software, and access to open data. However, for these new approaches to become sustainable, new approaches to collaboration R&D and commercialization are required. In this panel, three leaders practicing scientific R&D will discuss collaborative and business models for rapid innovation. Brian Wylie is a project lead at Sandia National Labs and a principle developer of the Titan informatics toolkit. Bill Hoffman is a developer of the CMake project and a major contributor to the ITK medical image analysis toolkit. Will Schroeder is CEO of Kitware, Inc. and developer of VTK, a popular open source visualization toolkit. These speakers will address the process of moving technology to practice using open source methods.

- **Cyber Situational Awareness**, U.S. Congressional Visitors, *CERL Open House*, February 2013 (Davis, Dunlavy, Rogers, Carvey)
  Presentation demonstrating Hybrid LDRD analysis capabilities associated with phishing e-mail detection and cyber situational awareness. Visitors included U.S. Senator Tom Udall and U.S. Congressman Ben Ray Lujan, along with staff and management from several corporations.

- **Cyber Situational Awareness**, *NM State Delegates from Finance Committee Visit*, June 2013 (Davis, Dunlavy, Oldfield, Carvey)
  Presentation demonstrating Hybrid LDRD analysis capabilities associated with phishing e-mail detection and cyber situational awareness. Visitors included several NM State Delegates along with their staff.

- **Hybrid Methods in Cybersecurity Analysis**, CIS EAB, FY12 (Wylie)
- **Hybrid Methods in Cybersecurity Analysis**, CIS EAB. FY13 (Davis)
- **Hybrid Methods for Cybersecurity Analysis**, FY13 Cybersecurity External Advisory Board (EAB) Meeting, 2013 (Dunlavy and Han)
- **The Hybrid Toolkit**, 2nd Annual Sandia Science and Technology Showcase, September 2013 (Davis)

3.5 Staff Participation

- Warren Davis (1400) — Principal Investigator, FY12-FY13; primary contributor, FY11
- Keith Vanderveen (8900) — Project Manager, FY11-FY13
- Danny Dunlavy (1400) — primary contributor, FY11-FY13
- Brian Wylie (1400) — primary contributor, FY11-FY12; Principal Investigator, FY11-FY12
- Christopher Nebergall (9300) — primary contributor, FY12-FY13
- Joe Ingram (9500) — framework and application developer, FY12-FY13
- Eunsil Han (9300) — analyst and security consultant, FY12-FY13
- Roger Suppona (9300) — analyst and security consultant, FY11-FY12
- Justin Doak (5600) — streaming analysis development, FY11
- Tim Shead (1400) — Splunk development, FY11
- J.T. McClain (1400) — framework and application developer, text analysis consultant, FY13
- Derek Trumbo (1400) — framework developer, FY13
- Nathan Fabian (1400) — framework developer, FY13
- Dave Dorsey (9300) — framework developer, FY12
- Andy Wilson (1400) — server development, text analysis, FY11-FY12
- Monzy Merza (9300) — Splunk development, data pipeline development, FY11
- Dave Robinson (1400) — text analysis, FY11
- Ted Reed (9312) – framework developer, integration, FY13
- Brad Carvey (1461 contractor) – visualization, capability demonstration, FY13

References

[1] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration*, Nov. 1999, pp. 229–238.

[2] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, Dec. 1999.

[3] B. Burns, E. Markham, C. Iezzoni, P. Biondi, J. Stisa Granick, S. Manzuik, P. Guersch, D. Killion, N. Beauchesne, E. Moret, J. Sobrier, and M. Lynn, *Security Power Tools*. O'Reilly, 2007.

[4] J. Wei, Y. Zhao, K. Jiang, R. Xie, and Y. Jin, "Analysis farm: A cloud-based scalable aggregation and query platform for network log analysis," in *Cloud and Service Computing (CSC), 2011 International Conference on*, 2011, pp. 354–359.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[6] T. Curk, J. Demsar, Q. Xu, G. Leban, U. Petrovic, I. Bratko, G. Shaulsky, and B. Zupan, "Microarray data mining with visual programming," *Bioinformatics*, vol. 21, no. 3, pp. 396–398, 2005.

[7] B. N. Wylie and J. Baumes, "A unified toolkit for information and scientific visualization." in *Visual Data Analytics (VDA)*, 2009.

[8] P. Nsholm, "Extracting data from nosql databases - a step towards interactive visual analysis of nosql data," Master's thesis, Department of Computer Science and Engineering (Chalmers), Chalmers University of Technology,, Jan. 2012.

[9] T. A. M. C. Thantriwatte and C. I. Keppetiyagama, "Nosql query processing system for wireless ad-hoc and sensor networks," in *Advances in ICT for Emerging Regions (ICTer), 2011 International Conference on*, Sep. 2011, pp. 78–82.

[10] E. Meijer and G. Bierman, "A co-relational model of data for large shared data banks," *Commun. ACM*, vol. 54, no. 4, pp. 49–58, Apr. 2011.

[11] R. Cattell, "Scalable SQL and NoSQL data stores," *SIGMOD Record*, vol. 39, no. 4, pp. 12–27, May 2011.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944937

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[14] J. C. Anderson, J. Lehnardt, and N. Slater, *CouchDB: The Definitive Guide Time to Relax*. O'Reilly Media, Inc., 2010.

[15] E. Plugge, T. Hawkins, and P. Membrey, *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing*. Apress, 2010.

[16] B. Wylie, D. Dunlavy, W. Davis, and J. Baumes, "Using NoSQL databases for streaming network analysis," in *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, 2012, pp. 121–124.

[17] B. Wylie, "Practicing open source." Presented at the O'Reilly Open Source Convention, Jul. 2011.

DISTRIBUTION:

1	MS 1318	Daniel M. Dunlavy, 1464
1	MS 1318	Warren L. Davis, IV, 1461
1	MS 0899	Technical Library, 4536
1	MS 0123	D. Chavez, LDRD Office, 1011

Sandia National Laboratories