Genoa TIE, Advanced Boundary Controller Experiment

Eric Monteith
NAI Labs, Network Associates Inc.
eric_monteith@nai.com

Abstract

This document describes experimentation performed as part of the Genoa Technology Integration Experiment (TIE). Achieved in two phases, the overarching assertion of the Genoa TIE was that boundary controllers, in the form of an automated guard, could play an important role in the operational success of Project Genoa [1]. Genoa, an ongoing DARPA [2] research program, is focused on developing a prototype decision support environment for the National Command Authority, and is intended to mitigate potential international crises early in their development. In addition to protection from Information Warfare attacks across the Internet and other sources, boundary controllers could assist the Genoa system in managing important aspects of information sharing, by implementing access control and content filtering for inter-enclave transactions. The focus of this paper includes experimentation with syntactic and Natural Language Processing filters within the Genoa environment, and the measurement of their effectiveness in filtering inter-enclave transactions.

Introduction

Genoa is focused on developing a prototype decision support environment for the National Command Authority (NCA) to identify and mitigate crisis situations early in their development. The organizations comprising the NCA span those of intelligence, operations, execution, and decision/policy control. Illustrated in Figure 1, each enclave comprising the NCA represents a homogeneous entity. They are often organizationally oriented and can be hierarchically organized. However, for any enclave to conduct the activities that it is responsible for, the staff in that enclave will have to make use of information and human resources external to that enclave. This information sharing and collaboration across enclave boundaries is essential functionality that Genoa will depend upon for fully attaining the program's goals.

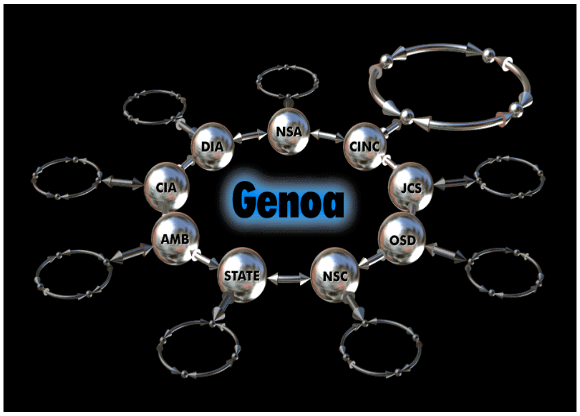

Figure 1: Conceptual architecture

Transactions that take place across enclave boundaries are of concern to each of the individual Genoa enclaves. Issues over need-to-know, proper examination of information, and release of information that could be interpreted as official position or policy, are of greatest concern. Consequently, the proper control of information flow across enclave boundaries is critical to the success of Genoa, and its acceptance in the existing organizational and process structures of the intelligence communities. By implementing boundary controllers with content-based filtering capabilities, locally controlled discretionary information sharing policies can be enforced, providing heightened security for Genoa enclaves and inter-enclave transactions.

Motivation

Many types of information are tightly held by organizations for a variety of reasons. Policies that restrict sharing may be based on any number of security reasons. When faced with requests to share information outside of the organization, someone within that organization must review the information to ensure that it may be shared without violating the organization's policy. Performed by security officers, this review can be both time consuming and error-prone. The primary goal of the Genoa TIE was to compare the accuracy and efficiency of the information sanitization and filtering process conducted by security officers, with automated filtering capabilities. To perform this comparison, we needed to establish an experimental data sharing policy, a collection of information resources, and a set of requests initiating the transactions. We also needed to develop the automated processes that would determine whether one organizations data could be released to another, based on policy.

The scenario used for the TIE was closely aligned with the FY-99 Genoa demonstration scenario and related data. That scenario involved the development of structured arguments that assess the capability and the intent of terrorist organizations to conduct chemical weapon attacks. The working example is that of the Aum Shinrikyo Japanese cult [3] that bombed the Japanese metro system with an anthrax pathogen.

Experiment design

Within the Genoa TIE, syntactic and Natural Language Processing (NLP) filters were investigated in an attempt to evaluate their effectiveness in terms of accuracy and performance against some baseline. A baseline was established through manual (human) content-based review of transaction data. We expected that each of the filtering methods (syntactic, NLP, manual) had varying strengths and weaknesses, and the strengths could be combined to achieve more accurate and efficient filtering results than any single filtering method. The goal of experimentation was to investigate ideal ways of combining automated, syntactic and NLP filters. Through this effort, three important aspects of our work became apparent. 1) We developed a process for measuring and comparing the content-based filtering capabilities of an automated guard. 2) We determined that syntactic and NLP filters, composed in a layered configuration, can provide better accuracy than either stand-alone solution. 3) While we found that our manual review provided the best accuracy, both of the automated filters were able to detect valid policy violations the humans had not detected. For the experimentation described within this document, the hypothesis was that the combination of syntactic and NLP filters in some configuration, would be better in terms of performance and accuracy than either filter method individually.

Experiment data

Data included information resources in the form of Critical Information Packages (CIPs), a set of requests for those CIPs, a security policy governing their releasability, and "meta-rules" that elaborated on how the rules were to be implemented.

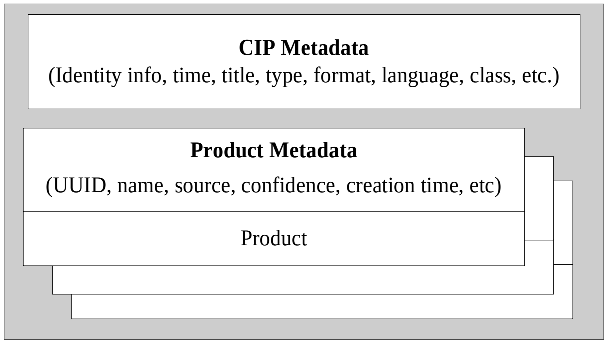
CIP Metadata
(Identity info, time, title, type, format, language, class, etc.)
Product Metadata
(UUID, name, source, confidence, creation time, etc)
Product

Figure 2: CIP structure

Created in Extensible Markup Language (XML), CIPs encapsulated a number of "products" and associated metadata (information pertaining to the product). Products were created from over 600 MB of raw data files obtained from the Genoa program that were converted into text, HTML, Word, Excel, PowerPoint, image, and Genoa Virtual Situation Book (VSB) files. Eighty-seven unique products were used to construct 34 unique CIPs with between 2-11 products each. This represents approximately 1200 policy-related metadata fields (author, source, description, creation date, time last modified, etc.). The products ranged in size from a small number of bytes to approximately 500K. The structure of a CIP is depicted in Figure 2.

Experiment security policy

The goal of the security policy was to represent an appropriate set of business rules for sharing data within a Genoa environment. Several CIP metadata fields were chosen for the filters to examine against the policy, including organization, sub-organization, user, author, CIP size, information type, information source, and information topic. Similarly, product metadata and product content were examined as well. The policy also included a set of "meta-rules" to provide baseline responses in the absence of more specific information and provide rule precedence in the case of conflicting rules. For experimentation, the policy was developed into an XML-like, Rules Markup Language (RML) that the automated filters could then interpret. Ultimately, 28 RML rules were developed for the experiment.

Filter technology

Two types of filters were implemented to perform content-based review of transaction data. Syntactic-only filtering was performed by Felt [4] filters developed specifically for the TIE, while NLP filter capabilities were provided by DataShield, a product developed by solutions-united [5]. These two filtering technologies are explained throughout the remainder of this section.

Felt was originally developed as a special-purpose language for developing filter procedures on guards, allowing a wide class of message formats to be characterized. Each time a Felt filter parses a message from its input, it applies the current policy to determine whether the message is releasable. In addition, the contents of a field may be altered (sanitized) in order to create a releasable version of a message. Historically, Felt had been used for filtering messages of the National Imagery Transmission Format (NITF), a complex format used for imagery products. The resulting filters were highly efficient yet provided detailed checking of the many header and sub-header fields contained in the messages. The Felt filters implemented within the Genoa TIE used a list of character strings, categorized under key-topic areas related to the theme of experimentation. While we had prior knowledge of the CIP data, this knowledge was not used to specifically develop our list of strings. For each of the key-topic areas of the experiment theme, "key-words" were chosen from related, open-source subject matter. This list of "key-words" was used to determine syntactic matches throughout the CIP meta, product, and product meta data.

NLP filtering performed by DataShield was much more complex than the syntactic only review. There are two distinct activities involved with implementing a trainable text classification system such as DataShield. The first is the actual training of the classifier. The second is its implementation in which it performs content filtering. The first of these tasks involves manually classifying a set of "training documents" in preparation for feeding them into the automatic system. Each training document is characterized as being "in" or "out" of range if it does or does not contain individual key-topic areas as outlined by their definitions.

The second step is to take these manually classified documents and process them with the trainable text classification system. During this process, it builds a "structure" of terms, phrases, and entities extracted from the text. Multi-level Natural Language Processing (MNLP) outputs are the basis for these textual data feature representations. The four types of analysis include:

1) Morphological analysis - words are stemmed to their root form.
2) Lexical analysis - words are tagged with their part-of-speech, type of noun, verb, and determiner.
3) Syntactic analysis - phrases identified; numeric concept, complex nominal, proper noun, non-compositional.
4) Semantic analysis - proper name interpretation; category of proper noun.

This collection of automatically generated features is then used to determine membership of new texts within a particular key-topic area. The DataShield system determines the "certainty of membership" for each of the documents as compared to each of the topic areas. Consider a range of 0.0 to 1.0, where 1.0 defines a document as containing a member of a certain class of key-topics and 0.0 defines a document as not containing a member of a certain class. Values of 0.0 and 1.0 both have a "certainty of membership" value of 1.0. This means that for either of these cases, it can be concluded with great certainty that the document either does or does not belong within a given class. If DataShield characterizes a document with a value close to 0.5, a "certainty of membership" value close to 0.0 will result. For these cases, DataShield cannot automatically determine whether a given document should be assigned to a given class. These documents are considered valuable in refining the classification system and its "knowledge base". By manually classifying these documents and then feeding them back into the automatic system, the system is trained to recognize the subtle differences that distinguish how these documents should be classified. During the CIP verification process, individual products are classed into predefined key-topic areas specified within the RML rules. Products that fall outside of these classes do not cause any of the rules to fire. If a product falls into one or more of the predefined classes within a given rule, and the other conditions of the rule have been met, the rule will fire with the specified action. For additional information about the NLP technology used within this effort, see [6].

Experiment baseline

Ultimately, in order to assess the accuracy and performance of the automated filters, a baseline for comparison was required. This baseline was obtained through manual, human review of the same transaction data that the automated filters processed. Our manual review was referred to as the Ground Truth (GT), and was intended to represent an ideal situation where all policy violations were detected throughout the CIP transactions.

Each of three human reviewers was provided with a package that replicated all transaction information, including all of the necessary meta, product, and environment data for determining the content-based releasability of each transaction. Environment data included all of the pertinent information needed for processing the CIPs with the same state as the automated filters. This information included the requesting user name and organization, time and date of request, CIP size, and key-topic definitions. The main goal for the baseline was to develop a “user-friendly” package for the human reviewers in an attempt to eliminate ambiguity and inconsistency in the review. The CIPs and policy were converted to HTML for ease of use, and provided with all the other pertinent data within a clearly labeled directory structure. The reviewers were also given a template in the form of an Excel spreadsheet to document their results. The reviewers examined all of the transaction data, searching for any policy violations while recording key-topic detections, rules fired, removal of CIP products, and the time it took to complete the review. Accomplished in parallel, the human reviewers produced very similar results, although there were differences stemming from key-topic detections and their context interpretation. When differences existed between the reviews, the results were inspected to verify the topic detection and context of the key-topic within the transaction data, and conflicts in interpretation were resolved among the reviewers. After completing this process of conflict resolution, one final GT document was established, representing the baseline for this experiment. While the GT was intended to represent an ideal assessment of the transaction data based upon the policy, we fully expected that the human review would not produce a completely accurate assessment. Fatigue, boredom, complexity of the task, etc., can affect the performance of human reviewers. While we later verified that the human baseline review omitted certain key-topic detections, it still provided a valuable basis for comparison in judging how well the syntactic and NLP filters performed in both accuracy and speed of review. It also provided evidence that it was possible for the automated filters to detect certain events that the collaborative, “ideal” human review had missed.

Experiment architecture

Based on the established goals, we simplified experimentation to a point where clear and concise accuracy and performance measurements could be collected, including a simplified network topology and configuration. The logical experiment topology depicted in Figure 3 describes the role of the NAI Labs Advanced Research Guard for Experimentation (ARGuE) [7] boundary controller within the Genoa environment. For each experiment run, 66 “transactions” were completed, each initiated with a request from the client to the CIP server. Requests included information necessary for accessing individual CIPs, including requesting user, user organization, target organization (location of CIP), and a unique CIP identifier. Once the client initiated a request, ARGuE received that request and executed the filters to assess the field contents. If the filters detected any policy violations for the request, that request was rejected. If the policy allowed, the request was passed on to the CIP server where it processed the request and attempted to return the CIP to the client. Again, ARGuE executed the filters on the reply, examining both the metadata and product content of the CIP being returned. If the policy allowed, the CIP was passed on to the client unmodified, completing the transaction. If policy violations were detected within the CIP, the CIP could be rejected or sanitized. An associated policy action

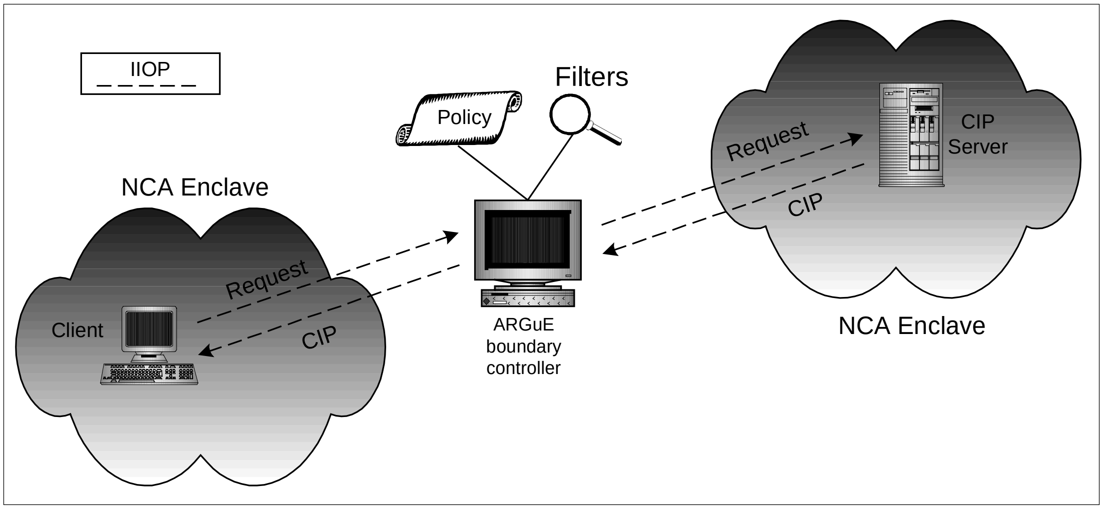

Figure 3: Experiment topology

allowed the filters to sanitize product metadata and/or remove products in order to satisfy the policy and allow release. This functionality allows the information flow to be controlled based on current operational risks and mission needs, instead of simply rejecting the transaction altogether.

Through the ARGuE filtering subsystem, all transaction results were logged for inspection. Through analysis of this data, we could determine the strengths and weaknesses of the various filtering methods. Experiment metrics were focused on the key-topics detected, the rules fired, and the products removed. Through additional manual review, these actions were then verified as being correct or incorrect. Overall, the experiment process included the following stages: 1) Establish experiment baseline that represents all of the policy violations throughout the transactions. 2) Perform automated filtering of the transactions and collect logging data for each filter. 3) Examine and document differences between the individual filter results and the baseline. The collection and analysis of this data enabled a better content-based filtering solution to be recognized.

Accuracy results

Accuracy statistics were collected for the Felt and DataShield filters by comparing their results with the GT assessment. Two factors were vital in obtaining a precise representation of the accuracy of the two filters. The first was the assumption that the GT was a sound representation of the actual CIP policy violations, and the second assumption was that the accuracy assessments were performed in the same manner for both filters. The GT assessment represents a human-in-the-loop analysis that was fully expected to fall short of being 100% correct in representing all of the violations contained within the CIPs. This provided opportunities for the automated filters to detect policy violations undetected by their human counterparts.

In gathering accuracy statistics, our approach included examining the profile of events within the GT assessment and comparing these to events recorded within the Felt and DataShield filter logs. Low-level analysis was achieved by examining key-topic detections resulting in product removal and/or sanitization events. After examining the logging data, product content violations were validated for correct key-topic areas as compared against the GT. If a disparity between the logging data and the GT existed, additional human analysis of the CIP and product was conducted. This allowed for determination of the correct action based on the policy and to double check the validity of the GT.

Spreadsheets were composed to capture the transaction activity of both Felt and DataShield filters. These spreadsheets were then summarized to provide specific differences in the two filters for each transaction, as compared against the GT. From these details, False Negative (FN), True Negative (TN), True Positive (TP), and False Positive (FP) statistics were gathered. These statistics were collected for product removal and product content detection for each of the filters. At the lowest level of analysis, product content detection statistics can be defined by the following:

FN: Content correctly identified by GT, but not identified by Felt or DataShield.
TN: Content incorrectly identified by GT and not identified by Felt or DataShield.
TP: Content correctly identified by Felt or DataShield, but not identified by GT.
FP: Content incorrectly identified by Felt or DataShield and not identified by GT.

Similar statistics were obtained for product removal, providing a rough accuracy measure without key-topic validation. While this does assess the product removal events of the filters against GT, it does not provide insight into why products were removed, and if they were removed for the correct key-topic detections. This metric is much easier to obtain, but does not provide a true assessment of accuracy.

In assessing the overall accuracy of the Felt and DataShield filters with these statistics, the Information Retrieval (IR) concept of Precision and Recall was utilized. Based upon our baseline GT, Precision corresponds to the ratio of false positive rule violations detected by the filters, while Recall corresponds to the ratio of false negative rule violations detected by the filters. The general formulas for Precision and Recall include:

$$\textbf{Precision} = \frac{\text{(\# of correctly identified items)}}{\text{(total \# of identified items)}}$$

$$\textbf{Recall} = \frac{\text{(\# of correctly identified items)}}{\text{(total possible \# of correct items)}}$$

For the calculation of these statistics, product removal and key-topic detection instances were extracted from the aforementioned spreadsheets. In addition, the total number of identified events detected was compiled from the GT assessment, denoted by ‘gt’ in the formulas below. The resultant Precision and

Recall formulas applicable to Felt and DataShield statistics result in the following four equations at the product removal and topic detection level of analysis.

$$\textbf{Precision}_{\text{(product removal)}} = [(gt-FN-TP) / (gt-FN-TP)+FP]_{\text{(product removal)}}$$

$$\textbf{Precision}_{\text{(topic detection)}} = [(gt-FN-TP) / (gt-FN-TP)+FP]_{\text{(topic detection)}}$$

$$\textbf{Recall}_{\text{(product removal)}} = [(gt-FN+TP) / (gt+TP)]_{\text{(product removal)}}$$

$$\textbf{Recall}_{\text{(topic detection)}} = [(gt-FN+TP) / (gt+TP)]_{\text{(topic detection)}}$$

Figure 4 summarizes the calculation of these statistics, providing Felt and DataShield Precision and Recall ratios for product removal (product) and topic detection (topic).

Ideally, precision and recall ratios of 100% are desired. From these statistics, it is apparent that Felt experienced a higher level of precision than that of DataShield, while DataShield experienced higher Information Recall results. Although Felt experienced only a slightly lower occurrence of false positive events than that of DataShield, DataShield was able to correctly identify key-topic areas, and sanitize or remove products according to policy with a significantly higher accuracy rate then Felt. Also visible from these statistics is the variance in Product and Topic level analysis, strengthening the fact that simply examining the product removal rate is not a good representation of accuracy where lower-level key-topic detail can be obtained.

Masked by these precision and recall ratios, is the fact that Felt and DataShield were both able to detect 41 key-topic instances that the human reviewers had missed. This strengthens the assertion that automated syntactic and semantic filtering not only supplement one another, but they also provide additional filtering accuracy beyond that of an “ideal” human review.

Performance results

To assess the performance of the Felt and DataShield filters, the timing data collected for the GT assessment provided the basis for comparison. The times collected by the human reviewers are approximate processing times for each transaction. These times, recorded in minutes, represent the duration of review of both the request and response. The corresponding transaction times for Felt and DataShield were extracted from the experiment logs, although recorded with finer granularity.

By utilizing timing log library routines, Felt was able to record timing measures in microseconds. DataShield implemented a general-purpose timing log interface that allowed for the inclusion of a time-stamp recorded only in seconds. In addition to the timing data generated by the two filters, the ARGuE filter subsystem also created timing information through the use of the general-purpose interface. The subsystem created each timestamp before and after the execution of the request and response filters. Thus, timing messages created by the filters are encapsulated within the filter subsystem messages. All of these possible timing methods offer varying levels of precision. Due to the incompatibility of the filter-generated timing data (seconds vs. microseconds), the subsystem timing data was used for comparison with the GT results. Because

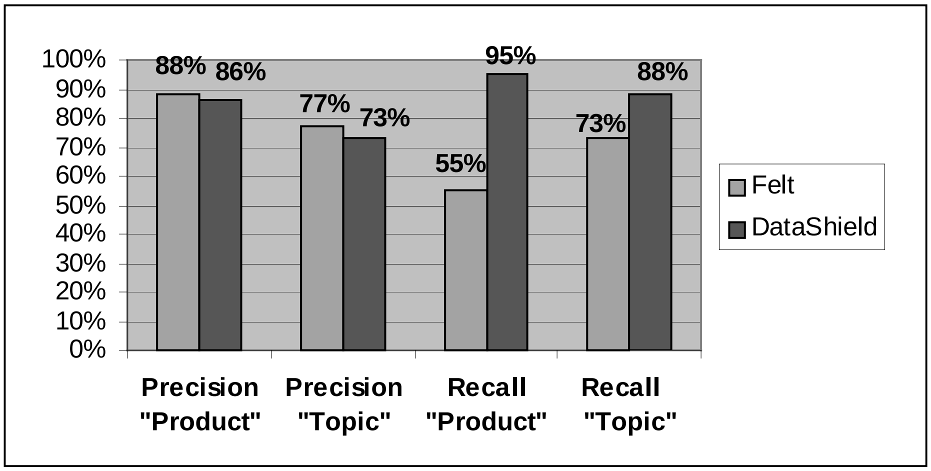

Figure 4: Precision and recall ratios

of this fact, the collected times for Felt and DataShield are slightly inflated (compared to timing data collected through the library routines measuring time in microseconds). While the subsystem timing data wasn't the most accurate measure, it does provide consistent measure of the overhead incurred for filtering the request and reply of each transaction.

Figure 5 provides a graphical view of the mean average transaction filtering times incurred for Felt, DataShield, and human review. All of the timing data is displayed in seconds and shown on a logarithmic scale. Due to size constraints, only odd numbered transactions are shown.

For each transaction, Felt outperformed both the DataShield and human review with markedly faster times. Except for transactions 30 and 39, DataShield outperformed the human review. These anomalies are explained through inspection of the CIP data, where transactions 29 and 30 as well as 38 and 39 contained the same products. The human review of transactions 30 and 39 were greatly reduced because of knowledge retained from the previous transactions. This same effect is visible through the gradual decrease in human review time for the first seven transactions, all of which contained the same products. Although in this case the assumptions made by the human reviewers were correct, this practice could have lead to incorrect assessments. Due to the mixture of sources used, there was no guarantee that the products contained the exact same data.

Conclusion

Human-in-the-loop content analysis is a resource intensive operation. Due to fatigue, boredom, and other factors, human reviews can be time consuming and error prone. Automated content filtering can supplement, and may eventually replace, manual content-based reviews as technology advances.

Through Genoa TIE efforts, we found that automated syntactic and NLP capabilities could be measured to determine the filters' strengths and weaknesses. Metrics were recorded in both accuracy and performance, and based upon a controlled human review. From these measures, current technology limitations were easily recognized and ideal configurations could be surmised based on tradeoffs in accuracy, performance, and risk. While our human review still provided the most accurate assessment, it is important to note that it represented an ideal situation,

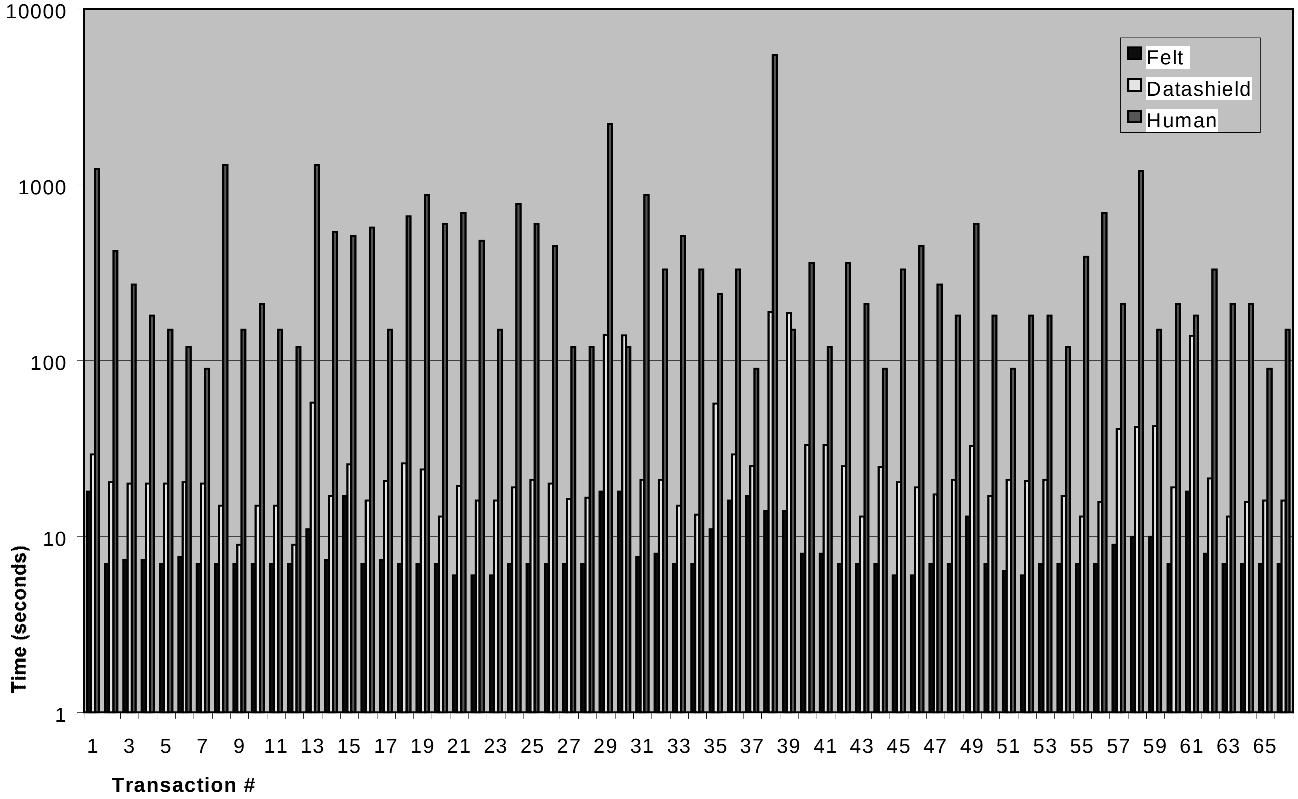

Figure 5: Total transaction times

with considerable processing effort and time expended by multiple reviewers.

In evaluating the content filtering abilities of Felt, DataShield, and the human review, findings confirm that no one content filtering method is a completely reliable solution. This fact is solidified by current day implementations of guards within true Multi-Level Security (MLS) environments. In most cases, automatic downgrading or sanitization within classified environments is not possible, because much of the data is unstructured. Other issues include the complexity and conveyance of the English language, and its numerous possibilities for interpretation. Transforming English policy into a portable, machine-readable format has proven a formidable task.

For the Genoa TIE, a very complex policy set was instrumented via RML. Although we determined that RML was not suited to support these complex rules as accurately as hoped, its use within the TIE was vital in collecting the presented data. Current guard implementations do not tend to institute such complex policies, and solutions for representing such complex, portable policies without room for interpretation, do not exist. Continued research in this area may prove useful in developing future solutions for next generation policy-based filters.

The Felt syntactic filtering system implemented within this experiment proved less accurate than expected. Although Felt experienced a low False Positive rate, it also experienced a low, positive identification rate for valid policy violations. Since the Felt filters rely solely on a keyword list of specified key-topic areas, its ability to perform well has to do with the careful selection of those keywords. The keywords for this experiment were gathered from various open source documents pertaining to the key-topic areas, and were not chosen from known CIP data. A more comprehensive key-word list may have provided better results, although False Positive rates would most likely have increased. Felt was considerably faster than either DataShield or human review, and Felt correctly identified key-word instances that DataShield and the human reviews both missed.

MNLP performed by DataShield was significantly better than the syntactic only review performed by Felt. While not as accurate as the human review, DataShield was considerably more efficient at processing the transactions. DataShield did suffer from a degree of False Positive detections, although it correctly identified most of the key-topic policy violations within the products. In terms of sensitive information transfer, we assert that it is certainly more desirable to erroneously withhold information that doesn't violate the policy than release information that does. However, an overprotective system can constrain the effectiveness of the mission at hand.

Overall, our findings support that the various filtering methods can be combined to provide a better filter configuration than any single solution. The NLP capabilities of DataShield certainly supplemented the abilities of the syntactic review, although each of the filtering methods detected content violations that the other methods did not. By combining the strengths of Felt syntactic filtering and DataShield NLP capabilities with manual review, increased levels of accuracy and efficiency could be obtained. Ideally, tradeoffs in accuracy, performance, and risk can result in an automated solution that is more desirable than manual review.

Within the Genoa environment, the implementation of ARGuE along with Felt and DataShield filters did provide the capability to perform access control among enclaves. Although several problems did exist with the correct implementation of the policy due to human misinterpretation, the system was able to accept an updated policy and to the best of the filters abilities, enforce that policy. Experiment data highlights the fact that Felt and DataShield were able to correctly detect all previously identified violations within the metadata. This is most likely due to the well-structured nature of the metadata where less ambiguity is involved. Because the syntactic review of the metadata was considerably faster and was just as accurate as the NLP review, Felt would be the better choice for filtering CIP and product metadata. DataShield is better suited for the unstructured product content, where it excelled at interpreting the meanings of the words contained within. By accepting some level of risk, an efficient, automated solution comprised of Felt and DataShield could provide Genoa with the necessary access control and content-based filtering of inter-enclave transactions. Within a high assurance environment, a hybrid of automated filters and manual processing could provide additional accuracy and increased efficiency to manual-only reviews.

Within this experiment, MNLP surpassed the detection capabilities of the syntactic filters. Could this technology be implemented within other security realms? For instance, current Intrusion Detection Systems (IDS) essentially filter network traffic for specific, known attack strings and sequences of events. Could similar technologies be "trained" to analyze traffic with a higher degree of accuracy, capable of detecting novel attacks?

References

[1] www.darpa.mil/iso2/project_genoa/project_genoa_white_paper.html

[2] www.darpa.mil

[3] www.gospelcom.net/apologeticsindex/a06.html

[4] J. Guttman, J. Ramsdell, and V. Swarup, *Felt: A Security Filter Compiler*, Personal Communication, November 1998.

[5] www.solutions-united.com

[6] www.solutions-united.com/products_technology.html

[7] J. Epstein, *Architecture and Concepts of the ARGuE Guard*, Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC), December 1999.