Data Leakage Prevention for Secure Cross-Domain Information Exchange

Kyrre Wahl Kongsgård, Nils Agne Nordbotten, Federico Mancini, Raymond Haakseth and Paal E. Engelstad

***Abstract*—Cross-domain information exchange is an increasingly important capability for conducting efficient and secure operations, both within coalitions and within single nations. A data guard is a common cross-domain sharing solution that inspects and validates that the security labels of exported data objects are such that they can be released according to policy. While we see that guard solutions can be implemented with high assurance, we find that obtaining an equivalent level of assurance in the correctness of the security labels easily becomes a hard problem in practical scenarios. Thus, a weakness of the guard-based solution is that there is often limited assurance in the correctness of the security labels. To mitigate this, guards make use of content checkers such as dirty word lists as a means for detecting mislabeled data.**

To improve the overall security of such cross-domain solutions we investigate more advanced content checkers based on the use of machine learning. Instead of relying on manually specified dirty word lists, we can build data-driven methods that automatically infer the words associated with classified content. However, care must be taken when constructing and deploying these methods as naive implementations are vulnerable to manipulation attacks. In order to provide a better context for performing classification, we monitor the incoming information flow and use the audit trail to construct controlled environments. The usefulness of said deployment scheme is demonstrated using a real collection of classified and unclassified documents.

I. Introduction

THE need for efficient information exchange within national armed forces, coalitions, and between military and civilian entities has received significant attention in recent years. This need is in strong contrast with the traditional approach to securing classified military systems, where isolation of security domains and information systems has been the default approach. Thus, concepts such as NATO's Information Exchange Gateways (IEGs), and similar initiatives within the nations, have been established to enable cross-domain information exchange in a secure manner.

These cross-domain solutions are required to perform various security controls, (e.g., information flow control, antivirus, and access control) to ensure that the interconnection does not compromise confidentiality, integrity, or availability. In addition, non-security specific requirements such as what type of information needs to be exchanged (e.g., friendly force identification, chat, or documents), and protocol specific details, may also impact security and what type of security controls are required. A key challenge, particularly when interconnecting domains at different classification levels, is to ensure sufficient assurance in the information flow control so that classified data is not leaked.

Solutions for collaboration and information sharing across security domains may generally be categorized as transfer solutions or access solutions. A transfer solution enables the transfer of information from one domain to another, while an access solution provides a user access to services and/or information within another domain without logically transferring the information from that domain. In the latter case the access solution itself may be viewed as an extension of the domain to be accessed, imposing the domain separation requirements on the access solution (e.g., a thin client connected by a secure tunnel). Transfer solutions may be further categorized based on their ability to provide one-way or two-way transfer. E.g., one-way data diodes are frequently used when information needs to be moved from a lower classified domain to a higher classified domain, while two-way information exchange may be enabled using a security filter or guard. We here use the term guard to refer to solutions basing their release decisions (at least partly) on security labels, while it may otherwise perform similar checks as a security filter (e.g., ensuring that data objects are according to some predefined format).

Assuming that security labels are correct, a guard may provide stronger security than a security filter alone, as a security filter typically may be bypassed by anyone knowing the allowed message format. This may to some extent be mitigated by having the security filter authenticate senders, but the use of security labels nevertheless provide an additional layer of security and also better applies to content whose sensitivity typically can not be determined by its format or type, such as documents, emails, or chat messages.

Before a user or service can initiate a request to export a data object, it must first be assigned a security label. This label is cryptographically bound to the data object. While the integrity of the data object and security label as such is cryptographically protected during transfer and storage, it is much more difficult to ensure that the correct security label is attached in the first place. For instance, if a RESTRICTED document is labelled as UNCLASSIFIED, it may result in it being released to an unclassified environment (i.e., leaked). Such mislabelling may be due to human or technical errors, or be due to users or malware trying to bypass security controls.

While the use of high assurance operating systems and applications may significantly reduce the risk of technical errors and malware, the use of commodity general purpose operating systems and applications are often mandated due to practical and economical reasons. This lack of assurance in end-user systems may in some cases be mitigated by labelling data objects based on origin, where a potentially high assurance intermediary mechanism (e.g., gateway) labels all data from a given origin (e.g., computer or network) with a given classification (e.g., RESTRICTED). However, this

approach would not allow documents from the same origin to have different security classifications. Thus, while applicable in some scenarios, this approach is often too inflexible to be practical. In the more general cases, the security label needs to be determined based on the content, rather than the origin, of the data object.

To mitigate the risk of incorrect security labels, another layer of protection in the form of a *content-checker* may be applied. For text-based data objects a "dirty word list" is often used, which scans the object for the presence of keywords that are often associated with classified content, e.g., security classifications, certain technical terms, locations, and project acronyms. The effectiveness of these content checkers are fully dependent on the quality of the rather static dirty word list in use. Given more recent advances in use of machine learning, data-driven content checkers based on machine learning have the potential to improve security of guard based cross-domain solutions.

This paper highlights our experiences in developing secure, scalable and robust cross-domain solutions (using data guards) and methods for increasing the assurance in the correctness of the user or application assigned security labels. Furthermore, it provides an in-depth view into the security challenges faced when using machine learning to create data-driven content-checkers for data leakage prevention (DLP).

II. Prototype High Assurance Guard

In cooperation with Thales Norway AS we have developed two prototype guard implementations, the first for use in service-oriented architecture (SOA) and the other to support cross-domain chat. The first guard [4] supports SOAP, which is an XML-based protocol for machine-to-machine communication, messages as used in Web services, while the chat guard [5] supports instant messaging through the Extensible Messaging and Presence Protocol (XMPP). Both guards are based on the core of a military messaging guard being developed by Thales Norway and target a Common Criteria EAL 5 certification. The guards are in alignment with the HAAG protection profile proposal [14] and uses the proposed STANAG 4774 for XML confidentiality label and STANAG 4778 for binding label and data.

Fundamental to the guard design is the use of a high assurance separation kernel. While many different guard implementations exist, most of these are based on medium assurance operating systems effectively preventing evaluation at higher assurance levels. The separation kernel ensures that different partitions (e.g., virtual machines or processes) cannot influence each other except by using well-defined interaction mechanisms. This allows security critical functions to be separated and protected from non-critical functionality and helps ensure least privilege and non-bypassability. Together the strong separation, high assurance, and ability to control communication between components (i.e., partitions) makes for a good environment to build high assurance systems.

Functionally the guard is separated into several different components, each implemented as one or more partitions. Central to the design is the core component which ensures that each object passed to the guard is processed correctly. This includes subjecting the object to label and signature checks, content checking and other access controls configured. Content checking is done through a separate component which provides a generic plug-in interface for content checkers. Depending on the scenario, different content checkers (e.g., malware scanning and/or format checking such as XML schema validation) can be included as needed. This architecture allows new content checkers to be added without risk of compromising other guard components.

Protocol adapters provide the interface towards the interconnected domains. Different protocol adapters are used to handle the specifics of a given protocol, e.g., XMPP or SOAP/HTTP. The main task of a protocol adapter is to extract protocol dependent attributes and transform these to protocol-independent attributes used by the core component. Additional components are used for handling configuration and the public key infrastructure. This architecture makes it easier to add new protocols without changing the security critical code of the core, and thus simplifies the certification process.

The guards' primary release control mechanism is label checking. A Mandatory Access Control (MAC) policy specifies the label ranges allowed to pass and how to handle unlabelled or incorrectly labelled objects. Other controls are also available for configuration, including allowed source and destination addresses, integrity control, and content-checkers. When multiple such controls are in effect, a message may be blocked from release if failing any one of these checks.

To support different applications and information exchange requirements, guards will have to handle messages and protocols of varying complexity. The functional needs must always be balanced against the need for security protection. Examples of this includes presence status, which in XMPP chat provides information about who is logged on and available. This information is very useful for the users, but can also be highly sensitive since it can reveal who is on duty and allow mapping of work schedules. Whether or not to allow this information to flow between security domains depend on the scenario and the level of risk acceptance. Support for this is given through configuration of the guard. Messages may also have different types of attachments, that may pose their own security risks and may require separate content checkers. Again, what is to be allowed needs to be determined be weighing the operational gain against the additional security risk.

The prototype guards are designed with high assurance certification in mind and the risk of information leakage due to compromise or malfunctioning of the guard is thus minimized. However, the trustworthiness of the primary release control mechanism, label control, is limited by the trustworthiness of the security labels themselves.

As of now, we do not have tools that can autonomously estimate the sensitivity of a text with a degree of confidence high enough to reduce the risk of information leakage to an acceptable level. Simple dirty word lists are quite limited. It is then natural to investigate the possibility of developing more effective automated classification techniques by using recent advancements in the field of machine learning. The second part of the article will be devoted to the latest research on this

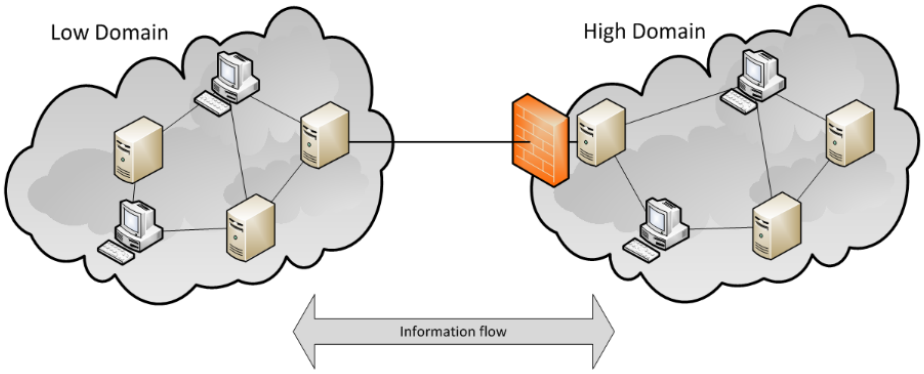

Fig. 1. The data guard enables two-way information flow between a "High" and "Low" domain. Each object passing through the guard will have its security label validated, its content checked according to policy/configuration (e.g., content may be scanned for malware and the presence of "dirty words"), and its sender and destination fields may be verified and subject to access control. Having passed these checks, the object is then released on the condition that its security label is such that it is considered, as specified by the governing security policy, to be releasable.

topic.

III. Machine Learning-Based Content Checking

Machine learning lends itself naturally to the problem of classifying unstructured textual information. However, much of the available research focuses on classification of text into a set of predefined topics, which is not directly applicable to our problem. The sensitivity of a text does not depend only on what it talks about, but also on the context in which it was produced and what kind of damage the information can do if leaked. This type of assessment is difficult even for a person, let alone for an algorithm.

As long as existing data has a known classification, it is possible to verify that it is not labeled incorrectly by employing direct comparison techniques like hashing. Estimating the sensitivity for completely new [3], [6], [15] or heavily processed (rewritten, summarized etc.) [8] information on the other hand is more challenging and is better handled using machine learning. By presenting the algorithm with examples of known classified and unclassified documents it will attempt to infer which features that are associated with each of the target classes (classification levels).

In order to further improve the performance, we worked on two ideas. The first consists in training the algorithm with an even more specific context to increase the accuracy rate. The other explores the possibility to improve the probability of detecting users (or other entities) with an abnormal amount of likely misclassifications over time, rather than aiming at detecting misclassification of single documents. In a guard setting, automatic classification may also be used to prioritize which documents are to be subject to manual review, in which case a somewhat lower classification accuracy may be acceptable.

In the remainder of this section, we will provide an overview of the challenges faced and the state of the art in developing and engineering machine learning-based content-checkers for the cross-domain information exchange setting.

A. Features

Before any learning can take place, the documents must be transformed into feature vectors. Feature engineering refers to the process of capturing an important characteristic of a document as a numerical value (feature). It is the part of the machine learning process that requires the most in-depth domain expertise, and is, together with the size/quality of the training data set and the choice of model class, what has the greatest impact on the performance of the resulting model.

Features for textual content are primarily derived from variations of word counts/frequencies, but one also uses more general features such as the average sentence length, the number of capitalized words and statistics regarding punctuation. Advanced features such as the part-of-speech (PoS) and named-entity recognition (NER) tags of words in a sentence are also beneficial for certain classes of tasks. A list of the features that we have used for the machine learning-based content-checker are:

- ***N-gram***: An n-gram is a contiguous sequence of n words. Term-frequency inverse document-frequency weights (TF-IDF) modifies these frequencies such as to better reflect the importance of a particular n-gram for the document. In the bag-of-words (BoW) model a document is represented as a multiset of its n-grams. While the BoW model discounts word order (expect for what is captured within the n-gram) and any grammar, it retains the semantic aspects and has been shown, despite its simplicity, to be very useful for text classification and in information retrieval systems [10]. N-gram frequencies can also be computed on the character level.
- ***Lemmatisation***: Lemmatisation is the process of grouping together the inflected forms of words, e.g., "flies" is mapped to "fly" and "better" is mapped to "good". This pre-processing step could be beneficial for sparser documents and for detecting paraphrased content and the use of synonyms.

Features can be extracted in parallel with their outputs concatenated into a single high-dimensional vector representation.

B. Controlled Environments

In a cross-domain scenario each data access request in the "High" domain can be logged on a per-user/session level. The audit trail of access requests, or the incoming information flow, can be used to derive what we have named controlled-environments. A controlled environment refers to any environment where we have control on all imported documents and their respective security classification. The set of imported documents, e.g., those accessed by the user during a session, is defined as *input*, and any new document(s) generated within the controlled environemnt is defined as *output*. By using the set of input documents as basis for the classifier, thereby reducing the noise in the classification process as the input documents are more relevant to the output, we can more accurately estimate the classification of output documents [8]. Our proposed solution inspects the information flow to the controlled environment as shown in Figure 2b, and estimates the classification of output documents based on the information about the input documents.

Experiments We want to analyze the performance for message-like (i.e. short) and using both a controlled environment setting as well as a traditional global classifier (one that uses the complete set of documents for training).

As a data set we use a subset of de-classified documents from the Digital National Security Archive. From this repository we extracted the three sub-collections:

1. *Afghanistan: The Making of U.S. Policy, 1973-1990*;
2. *China and the United States: From Hostility to Engagement 1960-1998* and
3. *The Philippines: U.S. Policy during the Marcos Years, 1965-1986.*

These were chosen because they contained a mix of both classified and unclassified documents from unrelated domains and from partially overlapping time periods.

We train the classifiers (l_2-regularized logistic regression) on documents, from the Digital Nation Security Archive (DNSA) data set [8], that were imported into a controlled environment and then evaluate the performance on the corresponding abstracts (these are removed from the input documents). This procedure simulates how (potentially classified) information is transformed into new documents. We have also studied other transformation models, e.g., the mixing of documents and the use of synonym (phrases), but we omit them from further discussions as the "abstract" transformation is the most realistic and challenging one. The leakage of known unmodified documents can be detected with very high accuracy (0.99) using both methods, and is not discussed further as this can be handled using existing methods (e.g., hashing).

In order to assess how well this methodology performs on short (e.g., message-like) documents, we use DNSA documents as the input/training data and evaluate it on sentence(s) sampled from the corresponding abstract. For comparison we also train global classifiers that use all the available data as a training set. In both cases we use the logistic regression implementation provided by the Python machine-learning library scikit-learn [12]. Cross-validation (5-fold) with a randomized hyperparameter search is used to determine the optimal value for the regularization coefficient, while features were extracted using the tool `TfidfTransformer` (tf-idf weights) from the scikit-learn package and the UDPipe pipeline toolkit (lemmatization) [13].

Figure 3 shows a visualization of how the classifier analyzes a document to determine its sensitivity level. The words highlighted in red indicates terms that are associated with the more sensitive class. For the particular example shown, it is clear that the model has learned that words such as "opposition", "nuclear", and "endanger" are often linked with sensitive information, while the terms "progress", "citizen", and "imprisonment" on the other hand are more likely to signify a non-classified document.

A comparison between a controlled environment and global deployment scenario is presented in Figure 4. It shows the accuracy of the model as a function of the number of sentences from the abstract that is sent as a message. Comparing the two graphs depicted, we see that we are able to achieve a significant boost in performance when using the per-user trained (i.e., controlled environment) model instead of the traditional global classifier. While this is a surprising result, and one that seemingly contradicts the conventional wisdom that more data always provides higher accuracy, it reflects that determining sensitive content is very context dependent and that we are exploiting the assumption that any classified content can be traced back to information contained in the imported documents. As such, a controlled environment would likely result in severe performance degradation if we wanted to use it to detect classified information that is completely unrelated to the input documents.

C. Internal Threat Scores

As the content-checkers are plagued with non-trivial false positive rates, we have also investigated the idea of constructing a meta-score called *Internal/Insider Threat Score* (ITS) that uses the aggregated confidence scores to detect long-term discrepancies between the user-assigned sensitivity level and the sensitivity level predicted by the machine learning model [9]. It works by modeling how the users (or other entities) assigns labels as a generative process and then infers (using a Bayesian network model) the latent variables that describe how often documents are misclassified in general for each user. These misclassification rates are what we use to compute the ITS. On a more technical level the model captures to which extent the deviations of the confidence score distribution for one user and the confidence score distribution for known classified documents can be attributed to incorrectly assigned labels by the user. By operating on a per-user (as opposed to per-document) level the number of false alarms is reduced. Figure 5 displays a visualization of the ITS.

Another concern that must be analyzed and addressed is the threat of the content-checker itself becoming a target for an attacker.

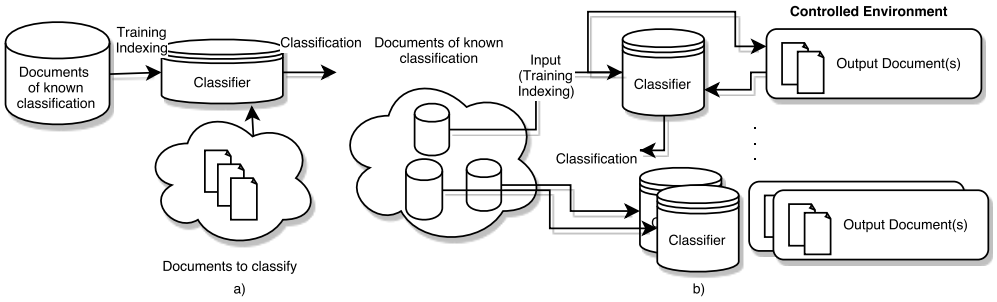

Fig. 2. **a**) Usually, a classifier used in DLP is trained on all available documents **b**) With a controlled environment, only the documents of known classification accessed from the environment are used to train the classifier, which in turn is used to classify documents generated within the environment. Multiple controlled environments can exist simultaneously, each characterized by its own input and output.

corazon c. (\ cory\ ) aquino reports no progress toward ending the aquino imprisonment (23 september 1972-8 may 1980); opposition leaders state that u.s. security assistance props up the marcos dictatorship (23 september 1972-16 june 1981); opposition groups will issue a manifesto against the presence of u.s. military facilities stating that the marcos dictatorship (23 september 1972-16 june 1981) is illegitimate and that nuclear weapons on the bases endanger philippine citizens

Fig. 3. Words highlighted in red are those associated with the classified content class, while green words are those associated with the unclassified content class. The darker the color the stronger the signal or the connection between the feature and the class is.

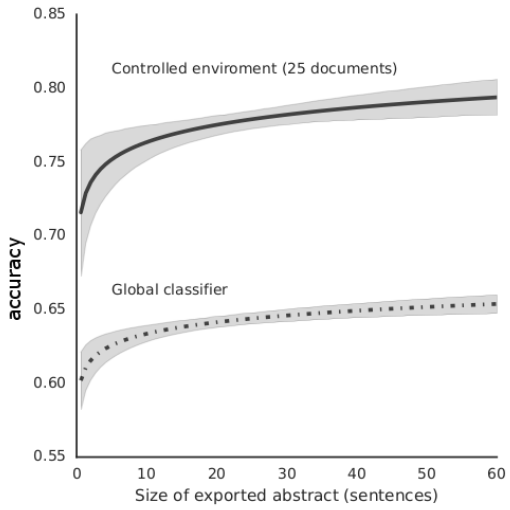

Fig. 4. Content-checker performance. Accuracy as a function of the number of sentences in the exported abstract sample, for both a controlled environment of size 25 and a global classifier using data derived from the DNSA dataset.

D. Secure Machine Learning

A core underlying principal behind most machine learning algorithms and tasks is that the training and evaluation datasets are generated from the same unknown distribution, i.e., it assumes a stationary environment. Under this assumption, minimizing the empirical risk (informally - the error) on the smaller training data set, which have often been painstakingly hand labeled, is equivalent with minimizing the risk on the larger evaluation data set. However, this assumption is violated for security tasks such as intrusion detection and DLP systems, where one must take into account the possibility of attackers

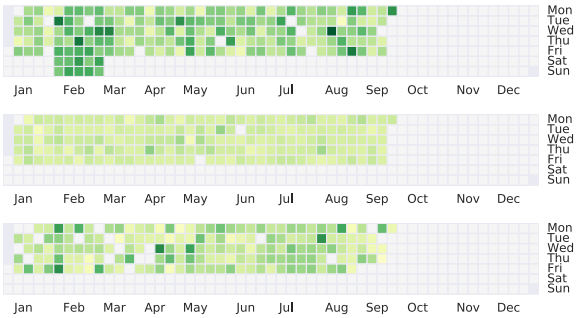

Fig. 5. A heatmap timeseries visualization of the daily ITS value (misclassification rate) for three users during a 9 month simulation period. A darker shade of green signifies a higher ITS value. **Top**: *A malicious user that has a very high baseline misclassification rate and periods of increased weekend activity.* **Middle**: *A regular user with a low misclassification rate.* **Bottom**: *Incompetent user with a high misclassification rate.*

actively seeking to by-pass detection by manipulating the classifier itself. A machine learning algorithm is said to be *secure* if it performs adequately when deployed in adversarial conditions.

Security assessments of machine learning systems is conducted with respect to the three axis [2]:

- ***Influence***: A user can influence the learning system by conducting either: a causative or an exploratory attack. Causative (interchangeably: poisoning) refers to manipulating (parts of) the training data with the intention of exerting control of the learning process. Exploratory refers to inducing and exploiting a misclassification, e.g., by rewriting a classified document such as not to trigger the content-checker.
- ***Security Violation***: Security violations takes on one of two forms: integrity, e.g., sensitive content being incorrectly classified and let through the guard, and availability, e.g., non-sensitive data being misclassified en masse, which may effectively render the system useless.
- ***Specificity***: The scope of the attack. It can be either a targeted or an indiscriminate attack.

An attack of particular concern in a content-checker context, is the possibility of data leaking from the model itself. If a user knows the functional form of the classifier (e.g., whether we are using a logistic regression, support vector machines or

some other model), and if the user can probe it for numerical outputs, i.e., if he has access to the probability/confidence scores for each data point, then through repeated experiments he can recover the actual parameter values of the underlying model or (parts of) data points in the training set [7]. When the training set contains classified information one must be particular wary of the possibility of the model leaking data.

We can analyze the security risks for the inferred model with respect to the attack categories/classes, estimate the feasibility of said methods, as measured in terms of the cost (risk and resources) incurred by the attacker, and propose potential mitigation steps:

- ***Exploratory*** (*Insider attacker*): A malicious user can, in theory, always bypass the detection mechanism by rewriting a document such as not to trigger the alarm. While this procedure can be automated for images [11], it remains a manual process for unstructured text. Taken to its extreme, we arrive at a scenario in which the insider employs methods of steganography to covertly embed classified information within other innocuous content. There does not exist a generic solution that solves this, and any solution must be combined with host-based systems to detect the presence of stenographic software.
- ***Causative*** (*Insider Attacker*, *Controlled Environment*): By carefully choosing what to import, the training set can potentially be shaped in such a way that the algorithm later misclassifies documents containing classified content that the user wants to exfiltrate. Defenses against these attacks include algorithms that effectively sanitize the data by modifying the learning process to dynamically discount those data points in the training set that have a significant negative impact on the performance. A competing class of defense mechanisms recasts the problem as one of anomaly detection, e.g., does the model parameters of one user deviate dramatically from the model parameters of other users. Similarly, performing a causative attack to exfiltrate larger amounts of data would likely result in detectable anomalies in the set of imported documents for a controlled environment setting.
  Instance-based algorithms (e.g. K-NN) are not as susceptible to causative attacks because: 1) there is no training phase involved and 2) we can use a decision strategy in which any imported document with a similarity score greater than a threshold value will result in assigning the strictest label, e.g., "Classified", to the document.
- ***Model Data Leakage***: When the model is invoked by the trusted guard there is no known way of performing such an attack as the user does not have access to the confidence scores. Furthermore, with a controlled-environment the user is already authorized to access the documents in the training set, which renders such an attack meaningless.

IV. Related work

The first work studying the use of machine learning to predict the security classification level of textual documents was done by Brown et al. [3] who built a binary classifier using only the abstract section of documents. Similar studies later reproduced and expanded upon these results [6], [15], [1] by using the complete document contents, multiple security-classification levels and per-paragraph sensitivity predictions, while the concept of controlled environments was introduced by us in [8].

V. Conclusion

In this paper we have discussed the use of high assurance data guards for controlling the information flow between different security domains.

We observe that one are currently able to develop guards, whose assurance level in practical scenarios surpasses the assurance provided by the security labels which the guard relies upon. Thus, while the guard only releases data objects with a security label releasable by policy, there is typically less confidence that those security labels are correct. Thus, more effective content checkers that detect such mislabeled data objects would be of significant value.

We have introduced the concept of applying machine learning techniques to construct automated, data-driven content checkers. Our treatment of the topic extends beyond the theoretical considerations by including what we have, through extensive experiments, observed to be the best practices for data-driven content checkers, including which features to use and how to deploy the classifiers. We have focused especially on assessing the impact different deployment settings have on the security and performance of the end system. As such, we have presented the concept of controlled environments, where the audit trail or incoming information flow is used to construct per-user/session classifiers, and which yielded a significant improvement in performance. The proposed methods have also been analyzed with respect to causative, exploratory and data leakage attacks, and we noted that while they still remain vulnerable to causative attacks carried out by sophisticated insiders, they completely alleviate the threat of the inferred model leaking sensitive data from the training set.

While previous work looked at building classifiers for complete documents [8], we have extended these methods to work for shorter messages, which is a more difficult case.

The performance we currently achieve is not sufficient to warrant a fully automated deployment scheme. However, with an appropriate decision threshold the classifiers can be used to determine which documents that must be manually assessed. A meta-score (ITS) operating on the per-user long-term classifications trends can also be used to further reduce and manage the number of false alarms [9]. As future work one can also investigate the feasibility and usefulness of other forms of classifiers, e.g,. language and genre detection for increasing the trust in exported documents. We have conducted preliminary experiments using an authorship verification model, built using the stylometric information embedded in the past chat messages of users, to detect instances in which an outgoing messages was not authored by the user in question. Combining the results from such different types of classifiers may potentially help improve accuracy.

REFERENCES

[1] Khudran Alzhrani, Ethan M Rudd, C Edward Chow, and Terrance E Boult. Automated us diplomatic cables security classification: Topic model pruning vs. classification based on clusters. *arXiv preprint arXiv:1703.02248*, 2017.

[2] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

[3] J David Brown and Daniel Charlebois. Security classification using automated learning (scale): Optimizing statistical natural language processing techniques to assign security labels to unstructured text. Technical report, DTIC Document, 2010.

[4] Raymond Haakseth, Nils Agne Nordbotten, Øyvind Jonsson, and Bengt Kristiansen. A high assurance guard for use in service-oriented architectures. In *International Conference on Military Communications and Information Systems*, 2015.

[5] Raymond Hakseth, Oddvar Brønstad, Øyvind Jonsson, Bengt Kristiansen, and Nils Agne Nordbotten. Cross domain communication using an XMPP chat guard. FFI-rapport 17/01491, Norwegian Defence Research Establishment (FFI), 2017.

[6] Hugo Hammer, Kyrre W. Kongsgård, Aleksander Bai, Anis Yazidi, Nils Agne Nordbotten, and Paal E. Engelstad. Automatic security classification by machine learning for cross-domain information exchange. In *Proc. IEEE Military Communications Conference*, volume 31, 2015.

[7] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.

[8] Kyrre Wahl Kongsgård, Nils Agne Nordbotten, Federico Mancini, and Paal E Engelstad. Data loss prevention based on text classification in controlled environments. In *Information Systems Security*, pages 131–150. Springer, 2016.

[9] Kyrre Wahl Kongsgård, Nils Agne Nordbotten, Federico Mancini, and Paal E Engelstad. An internal/insider threat score for data loss prevention and detection. In *International Workshop on Security And Privacy Analytics*. ACM, 2017.

[10] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[11] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[13] Milan Straka, Jan Hajic, and Jana Straková. Ud-pipe: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016.

[14] Konrad Wrona and Nadja Menz. Protection profile for the NATO high assurance ABAC guard (HAAG), version 1.3. NCIA Technical Report TR-2012-SPW0084-18-13-4, NATO Communications and Information Agency (NCIA), 2013.

[15] Konrad Wrona, Sander Oudkerk, Alessandro Armando, Silvio Ranise, Riccardo Traverso, Lisa Ferrari, and Richard McEvoy. Assisted content-based labelling and classification of documents. In *Military Communications and Information Systems (ICMCIS), 2016 International Conference on*, pages 1–7. IEEE, 2016.

Kyrre Wahl Kongsgård (kyrre-wahl.kongsgard@ffi.no) is a final year graduate student (Ph.D. degree) at the Department of Technology Systems, University of Oslo and a scientist at the Norwegian Defense Research Establishment (FFI). He received his M.Sc. degree in computational science from the University of Amsterdam (UvA) in 2013. His current research is concerned primarily with applying machine learning techniques to information security problems such as DLP and network intrusion detection systems.

Nils Agne Nordbotten (nils.nordbotten@ffi.no) is a research manager and principal scientist within ICT- and cybersecurity at the Norwegian Defence Research Establishment (FFI), and an adjunct associate professor at the University of Oslo (UiO). He received his Ph.D. (2008) and Cand.scient. (2003) in computer science from UiO, and an executive master of management degree (2012) from BI Norwegian Business School.

Federico Mancini (federico.mancini@ffi.no) received the M.Sc. degree in Computer Science in 2004 at "Università degli Studi Roma Tre", Rome, Italy, and the Ph.D. degree in Algorithms and Graph Theory in 2008 at the University of Bergen (UiB), Bergen, Norway. He has since then shifted his research focus to information security and is currently a Senior Scientist at the Norwegian Defence Research Establishment (FFI). He was also Adjunct Assistant Professor at UiB from 2011 to 2015.

Raymond Haakseth (raymond.haakseth@ffi.no) is a research manager and senior scientist within ICT and cybersecurity at the Norwegian Defence Research Establishment (FFI), where he has worked since 2004. He received his M.Sc. degree in computer science from the University of Tromsø (UiT) in 2003. His research interests include information assurance, distributed systems and software engineering.

Paal E. Engelstad (paal.engelstad@ffi.no) received his PhD in computer science from University of Oslo (UiO) in 2005. He now working as an research scientist at FFI, an adjunct professor at UiO and vice-dean and full professor at Oslo and Akershus University College (HiOA). His current research interests include fixed, wireless and ad hoc networking, cybersecurity, machine learning and distributed, autonomous systems.