CMU-AM-17-036

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Improving Security in Software Acquisition with Data Retention Specifications

30 March 2017

Dr. Daniel Smullen

Dr. Travis Breaux

Institute for Software Research
Carnegie Mellon University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-16-1-0006. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Abstract

The Department of Defense (DoD) Risk Management Framework (RMF) for IT systems is aligned with the National Institute for Standards and Technology (NIST) guidance for federal IT architectures, including emergent mobile and cloud-based platforms. This guidance serves as a prescriptive lifecycle for IT engineers to recognize, understand, and mitigate security risks. However, integrators are left with the challenge - during acquisition, and during runtime integration with external services - to reason about the actions on data inherent in their system designs that may have confidentiality risks. These risks may lead to data spills; loss of confidentiality for mission data, and/or revelations about private data related to service members and their families. Solutions are needed to assist acquisition professionals to align system data practices with the RMF and NIST guidance, as well as DoD IA directives - particularly with respect to the collection, usage, transfer, and retention of data. To provide support to this end, we extended our initial automation framework, to support reasoning over data retention actions using a formal language. We propose an evaluation method for these extensions, carried out through simulations of real-world IT systems using imitation but statistically accurate synthetic data. Our language aims to address dynamically composable, multi-party systems that preserve security properties and address incipient data privacy concerns. Software developers and certification authorities can use these profiles expressed in first-order logic with an inference engine to advance the RMF, express data retention actions that promote confidentiality, and re-evaluate risk mitigation and compliance as IT systems evolve over time.

THIS PAGE INTENTIONALLY LEFT BLANK

About the Authors

Daniel Smullen is a research assistant enrolled in the software engineering PhD program at Carnegie Mellon University. His research interests include privacy, security, software architecture, and regulatory compliance. [dsmullen@cs.cmu.edu]

Travis Breaux is an assistant professor of computer science in the Institute for Software Research at Carnegie Mellon University (CMU). His research program searches for new methods and tools for developing correct software specifications and ensuring that software systems conform to those specifications in a transparent, reliable, and trustworthy manner. This includes compliance with privacy and security regulations, standards, and policies. Dr. Breaux is the Director of the CMU Requirements Engineering Lab; co-founder of the Requirements Engineering and Law Workshop, and has several publications in ACM- and IEEE-sponsored journals and conference proceedings. [breaux@cs.cmu.edu]

THIS PAGE INTENTIONALLY LEFT BLANK

CMU-AM-17-036

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Improving Security in Software Acquisition with Data Retention Specifications

30 March 2017

Dr. Daniel Smullen

Dr. Travis Breaux

Institute for Software Research
Carnegie Mellon University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-16-1-0006. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

THIS PAGE LEFT INTENTIONALLY BLANK

Introduction

Service-oriented and cloud-based system architectures are becoming increasingly pervasive in web, mobile and desktop-based applications, both in the commercial sector, and within DoD IT acquisition roadmaps (United States Department of Veterans Affairs, 2014). This is in part motivated by the low-cost of network bandwidth across hardware platforms, and the availability and low-cost of remote commercial storage, such as GovCloud (Diez & Silva, 2013). Low cost and pervasive infrastructure, coupled with DevOps, Agile development (e.g., rapid build, test and release cycles), provide new opportunities to rapidly evolve systems. The culture of rapid and dynamic software development also introduces privacy and security challenges, specifically with respect to confidentiality and data retention. This makes commercial solutions acquisition and outsourcing to third party service-oriented architectures difficult. The variety and longitudinal nature of information collection introduces privacy and security risks that are especially difficult to predict, due to requirements creep brought on by emergent data spill risks and cyberattacks (Defense Science Board, 2013). These risks include hostile adversaries re-identifying individuals from anonymized or de-identified data, and/or inferring confidential attributes related to personnel data (e.g. health records, off-base addresses and contact information, information regarding family of service members), leading to loss of confidentiality and data spills. While mitigating technologies exist to reduce these risks under certain assumptions, the distributed nature of software makes it difficult to reason about and impose these requirements on information systems, especially when they involve data sharing and use by third-party services.

In the private sector, commercial data-driven innovation exposes the public to increased confidentiality risk, and this has direct impact for DoD when leveraging or interacting with commercial solutions for data management. Whereas emerging commercial data-driven practices frequently employ long term surveillance of customers to "personalize" services to those customers, defense customers may wish to reduce or limit this surveillance to avoid exposing defense applications to increase risk. For example, U.S. Transportation Command (TRANSCOM) learned in 2014 that multiple third-party, commercial services, including commercial airlines, information technology and shipping companies, were subject to advanced persistent cyber threats (Federal Bureau of Investigation, 2014). While commercial companies routinely treat and store all business records in the same manner, defense contractors are the target of more advanced cyber threats. In this respect, restrictions on how sensitive defense data is stored and shared could reduce the risk of using commercial services.

In this paper, we describe extensions to the Eddy requirements specification language (Breaux, Hibshi, & Rao, Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements, 2014; Breaux, Smullen, & Hibshi, Detecting repurposing and over-collection in multi-party privacy requirements specifications, 2015) to express and reason about data retention requirements. These requirements embody three strategies affecting data retention: *redaction*, which is the removal of elements from a data set; *data append*, which serves to link a data set to another data set, often to increase information about a data subject; and *perturbation*, which is the summarization of data using irreversible, statistical methods. We claim that these extensions to Eddy can be used to analyze trade-offs between data utility and data confidentiality. The extensions are coincident to existing *data minimization* strategies to reduce exposure (Ross, 2012), that includes simply not sharing data at all, sharing only the minimum necessary, sharing data but assuring disjointness downstream, and sharing only safeguarded data.

Running Example

We illustrate our approach using a running example wherein defense contractors must share cyber threat information with each other to provide a holistic view of security risk (see Figure 1). To be effective, contractors must collect and share information about their networks and affected data, including data about proprietary technologies that may have been stolen in a cyber-attack. This information is highly confidential and redacted to avoid revealing personal, proprietary, or otherwise secret data. Moreover, as this information is appended with other data, the combination can reveal sensitive information. Increasingly, effectiveness of threat information sharing requires maximizing confidentiality in light of the sensitivity. The Defense Industrial Base (DIB) works with the DoD Cyber Crime Center, Defense Security Service, Department of Homeland Security Enhanced Cybersecurity Services, and the Defense Security Information Exchange in order to run a cyber threat information sharing portal. This portal relies on third party contractors who are involved in the DIB to provide security incident information recovered from threats, and/or information about newly discovered threats, and is the subject of interest for our running example. The goal of this portal in our scenario is focused on one particular interaction between a cyber threat clearinghouse (CTC) and a third party contractor, in order to highlight the specific nuances of the confidentiality risks inherent in unanticipated disclosure of threat information. This interaction is specified by a fictional data sharing agreement wherein the CTC provides the contractor with information about known threats in order for the contractor to harden their systems against a novel attack (see Figure 2). Conversely, the contractor has collected data about the attack as if they have been subjected to it, but insufficient information is available for them to protect against further such attacks.

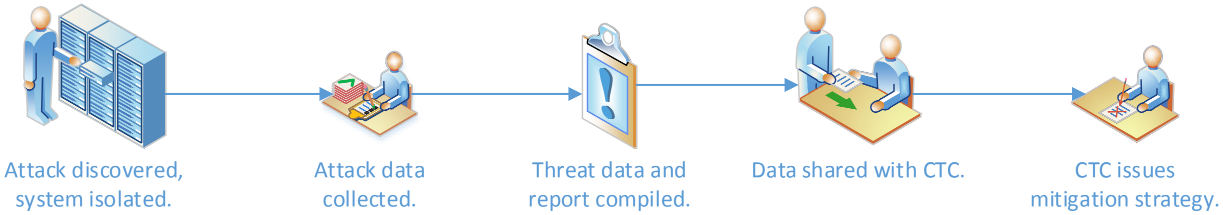
Attack discovered, system isolated.
Attack data collected.
Threat data and report compiled.
Data shared with CTC.
CTC issues mitigation strategy.

Figure 1 - Cyber threat information sharing process illustration.

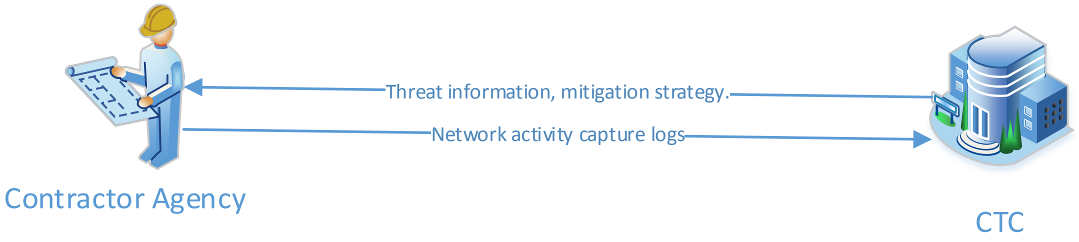
Threat information, mitigation strategy.
Network activity capture logs
Contractor Agency
CTC

Figure 2 - Cyber threat information sharing data flow illustration.

The CTC wants to help the contractor make their systems resilient to the specific threat they have encountered, but requires data about the threat in order to identify it and provide the necessary mitigation information. In order to do this, their threat information portal aggregates information about many attacks in a database. For each attack, the CTC assigns a database record to each attack that contains a unique identifier, associated organizations and adversaries known to employ the attack (which is classified intelligence), information about individuals known to be associates to these organizations (also confidential), and information which serves to mitigate the threat (which may be classified). The data sharing agreement requires that the contractor provide the data they have collected about the attack, which includes network activity capture logs. Such logs can contain confidential and proprietary information, specifically including identifiers for users within their organization, communications between users, technical information about the structure and composition of their network, in addition to evidence of the actual cyber threat. Thus, aside from the needed information, ancillary information would be present that would need to be segmented away from the threat information. The data sharing agreement would rely on data retention strategies to determine where and how separation and segmentation is achieved.

Technical Approach

The Eddy Requirements Specification Language

The Eddy language has formal semantics expressed in Description Logic (DL) - a subset of first-order logic for expressing knowledge (Breaux, Hibshi, & Rao, Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements, 2014). A DL knowledge base *KB* is comprised of intensional knowledge, which consists of concepts and roles (terminology) in the TBox *T*, and extensional knowledge, which consists of properties, objects and individuals (assertions) in the ABox (Baader, Calvenese, & McGuiness, 2003). In this paper, we use the DL family ALC, which includes logical constructors for union, intersection, negation, and full existential qualifiers over roles. Concept satisfiability, concept subsumption and ABox consistency in ALC are PSPACE-complete (Baader, Calvenese, & McGuiness, 2003).

Description Logic includes axioms for subsumption, disjointness, and equivalence with respect to a TBox. Subsumption describes individuals using generalities: we say a concept *C* subsumes a concept *D*, written $T \vDash D \sqsubseteq C$, if $D^{\mathfrak{I}} \subseteq C^{\mathfrak{I}}$ for all interpretations $\mathfrak{I}$ that satisfy the TBox *T*. The concept *C* is disjoint from a concept *D*, written $T \vDash D \sqcap C \rightarrow \perp$, if $D^{\mathfrak{I}} \cap C^{\mathfrak{I}} = \emptyset$ for all interpretations $\mathfrak{I}$ that satisfy the TBox *T*. Finally, the concept *C* is equivalent to a concept D, written $T \vDash C \equiv D$, if $C^{\mathfrak{I}} = D^{\mathfrak{I}}$ for all interpretations $\mathfrak{I}$ that satisfy the TBox *T*.

The universe of discourse consists of the set *Req* of requirements, *Action* of actions, *Actor* of actors, *Datum* of data types, and *Purpose* of data purposes. A specification is a DL knowledgebase KB that consists of multiple requirements. A *requirement* is a DL equivalence axiom $r \in Req$ that is comprised of the DL intersection of an action concept $a \in Action$ and a role expression that consists of the DL intersection of roles $\exists R_1 \sqcap \ldots \exists R_n \in Roles$. We are primarily concerned with four roles in this paper: *hasSource* indicates the source actor from whom the data was collected; *hasObject* indicates the data on which an action is performed; *hasPurpose* indicates the purpose for which data is acted upon; and *hasTarget* indicates the recipient to whom data is transferred. For example, requirement p_0 for a $location \in Datum$, and purpose $providing\_services \in Purpose$ in the TBox *T*, such that it is true that:

$$(1)\ T \vDash p_0 \equiv COLLECT \sqcap \exists hasObject.ip\_address \sqcap \exists hasSource.contractor \sqcap \exists hasPurpose.identify\_threat$$

Each requirement is contained in exactly one modality concept in the TBox *T* as follows: *Permission* contains all actions that an actor is permitted to perform; *Obligation* contains all actions that an actor is required to perform; and *Prohibition* contains all actions that an actor is prohibited from performing. We adapt the axioms of Deontic Logic (Horty, 1993), such that it is true that $T \vDash Obligation \sqsubseteq Permission$, wherein each required action is necessarily permitted. If the requirement p_0 is required such that $T \vDash p_0 \sqsubseteq Obligation$, then $T \vDash p_0 \sqsubseteq Permission$. We can now compare the interpretations of two requirements based on the role fillers to precisely infer conflicts. A *conflict* is defined as $Conflict \equiv Permission\ \sqcap Prohibition$.

Further Extending Eddy for Data Retention

The Eddy language syntax and semantics were extended to support reasoning over the data retention strategies: redaction, data append, and perturbation. Specifications written in the Eddy language to express these actions are called *profiles*. The data sharing agreement, and the data retention actions that are expressed in the agreement, are expressed in a profile.

Redaction means to remove data elements from a dataset. For example, we define information types captured from a network in formula (1):

$$(1)\quad T \vDash ip\_address,\ site\_location,\ employee\_id\ \sqsubseteq network\_captured\_information$$

The following Eddy permission (the P indicates permission) states that a new concept, $personal\_info\_redacted$ is equivalent to the interpretation of personal information excluding the interpretation of employee id numbers, such that formula (2) is true:

P REDACT employee_id FROM network_captured_information YIELDS net_capture_redacted

(2) $T \vDash \mathit{net\_capture\_redacted} \equiv \mathit{network\_captured\_information} \setminus \mathit{employee\_id}$

Redaction is useful as a data minimization strategy when data cannot be shared, or when it can be pared down to the minimum necessary.

Data append refers to a general class of methods that link two or more data elements together. For example, using a person’s e-mail address to link their organization with their IP address. By prohibiting data append, downstream parties in a data sharing agreement are bound to limit the use of a redacted dataset for the purpose of re-identifying individuals in otherwise de-identified data. This represents a fixed requirement for the data prior to sharing, assuring disjointness from other datasets post-transfer to a third party.

P APPEND organization_name,email_address,ip_address TO network_capture_information YIELDS net_capture_extended

(3) $T \vDash \mathit{net\_capture\_extended} \equiv \mathit{network\_capture\_information} \sqcup \mathit{organization\_name} \sqcup \mathit{email\_address} \sqcup \mathit{ip\_address}$

Perturbation refers to a general class of methods that introduces statistical inaccuracies into data (e.g., changing data values, removing values or adding new values that conform to a statistical profile). These inaccuracies are introduced to protect the confidentiality of individuals or data attributes in the data set, while ensuring the dataset can be used to obtain statistically accurate samples. In our running example, we employ statistical noise based on Laplacian distributions (as proposed by Dwork), which is a method that ensures that the presence or absence of an individual datum will not significantly affect the output of query over a data set containing that datum (Dwork, 2006). In general, the Eddy language does not assume that data perturbation is implemented by any particular method and is expressed as follows:

P PERTURB geolocation_record YIELDS geolocation_perturbed

The modalities in Eddy are applicable to all three data retention strategies, which means redaction, append and perturbation can be permitted, required or prohibited.

Experimental Design

The experimental design described herein aims to answer the following research question*: how do data append, redaction and perturbation systemically affect data subject unanticipated disclosure?*

To answer this question, we designed a *microsimulation*, which is a technique for analyzing real-world situations based on synthetic data, called microdata (Lovlace, 2016). Synthetic datasets consist of artificially generated data that satisfies real-world statistical distributions, and they are used to conduct experiments when publicly available data sets are unavailable (perhaps due to confidentiality reasons) or are too difficult to acquire. Highly specialized microsimulations have been used in a wide variety of domains, such as to predict the impact of social policy changes (Lovlace, 2016), explore portfolio affordability in carrier battle group designs (Vascik, Ross, & Rhodes, 2015), and various other applications in Homeland Security (Stamber, Brown, Pless, & Berscheid, 2013), used by the National Infrastructure Simulation and Analysis Center across a wide gamut of infrastructure and security concerns.

The inputs to the experiment consist of a formal Eddy specification containing data retention actions constrained by the scenario, synthetic datasets for each scenario actor (*CTC*, *contractor*), and our evaluation functions to predict data utility, specifically threat identification and mitigation probability, which are in part determined by expert analysis (see Figure 3).

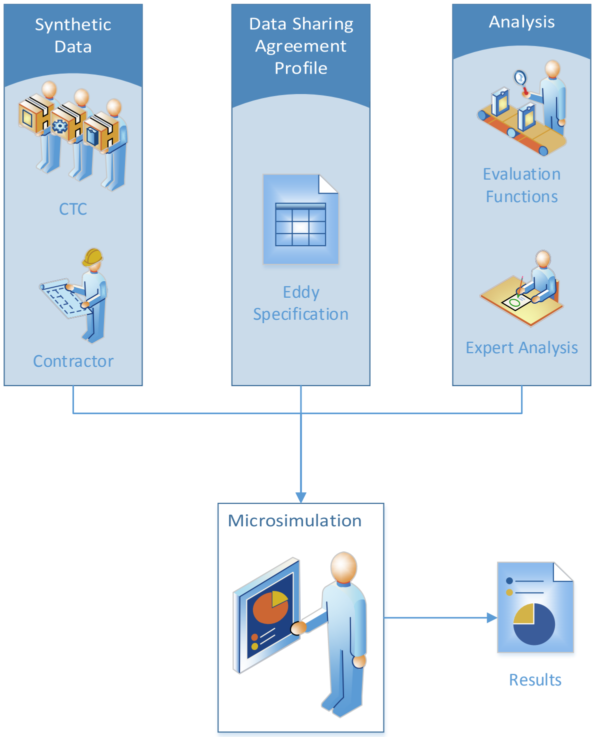

Figure 3 - Experiment Design illustration, showing the microsimulation inputs.

Sampling Data Retention Specifications

The microsimulation is based on randomly sampling from the population of data retention agreements expressed in the extended Eddy. We now describe our assumptions to design a minimal population of profiles limited to the threat information sharing scenario. In the scenario, there are two classes of profiles, one for each of only two data processors, the *CTC* and *contractor*. We assume both data processors use the same terminology[1] to describe the shared data, and each profile class will describe an additional, minimal set of data to meet each actor's respective business needs. Finally, each profile class includes a minimal set of rules to describe: (a) what data is collected from data subjects, the customers and patients; (b) what data is collected and shared between the data processors; and (c) how data is redacted, appended and perturbed by either processor. The set of rules in (a)

[1] See (Breaux, Smullen, & Hibshi, Detecting repurposing and over-collection in multi-party privacy requirements specifications, 2015) for extension to Eddy that accounts for different terminology between two or more actors.

above is fixed for both classes of profiles, and we let the rules in (b) above vary to equivocate with the data needed to answer the queries necessary to segment data, determine the threat, and provide the appropriate mitigation response.

For the rules in (c) above, we sample from all possible permutations of these actions. Because each action “yields” a new datum, the number of permutations of redaction, append and perturbation actions is infinite. However, the finite ontology limits the infinite space to a finite set of equivalence classes. Figure 4 presents a generic example in which three concepts A, B and C exist in two subsumption relationships in an ontology, indicated by black, solid arrows that point from sub classes to super classes: $B \sqsubseteq A$ and $C \sqsubseteq B$. The red arrows along the top illustrate redaction actions, e.g., rule R_1 yields a new concept D by excluding the interpretation of concept B from concept A. The purple arrows show the inverse, e.g., rule A_2 shows an append action that yields concept B by adding back the concept C to the redacted concept E. Thus, the rules R_1 and A_2 are part of an equivalence class described by the concept B and reachable by actions R_1 and A_2. We can envision an infinite number of such actions to reach each permutation of concept inclusion and exclusion, which comprises a finite set. Furthermore, we can perturb any concept permutation.

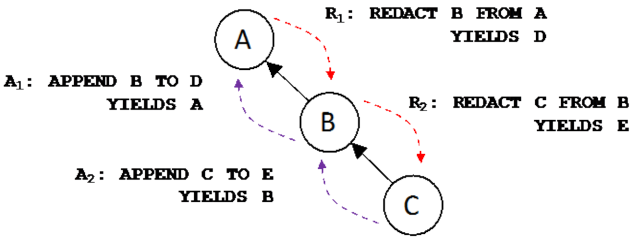
A
B
C
R_1: REDACT B FROM A YIELDS D
A_1: APPEND B TO D YIELDS A
R_2: REDACT C FROM B YIELDS E
A_2: APPEND C TO E YIELDS B

Figure 4 - Example illustrating equivalence classes for data retention actions over concepts in the ontology.

To simplify sampling, however, we only consider simple actions that modify original ontology concepts and that do not modify derivative concepts (in Figure 4, the derivative concepts are D and E, whereas A, B and C are original concepts).

Synthetic Data Generation

Our technical approach relies on both data processors using the same terminology to describe the shared data, and each policy class describing a minimal set of data to meet each actor's respective business needs. Thus, the synthetic data which is generated for this microsimulation is instantiating this data set. Microsimulations utilize aggregate data derived from surveys as a basis for producing microdata. The microdata matches the original statistical distribution of the aggregate data, but the individuals in the microdata may have different combinations of attributes that do not match individuals present in the original survey data. Thus, while each individual record in microdata is not a real record, the statistical profile of any guaranteed collection of records conforms to and generalizes to the statistical estimations in the aggregate data.

We propose using Monte Carlo (Mooney, 1997) to generate initial individuals in our microdata using reliable, aggregate data collected from security experts, and unclassified aggregate information from the CTC. Next, we introduce additional data elements to satisfy the level of detail in the model required to match our scenario. This data would imitate the classified data which would not be included in our simulation, and would be generated to match the model using purely fictional data. This data would include simulated attacks based on realistic adversaries and known attacks.

Evaluation Functions

Once synthetic data is generated, this data must be combined with the sampled profile. Each profile varies to equivocate with the data needed to answer the queries necessary to segment data, determine the threat, and provide the appropriate mitigation response, and the evaluation functions substantiate these queries.

To segment the data, expert analysis is required to examine the captured data and determine what is associated with threats, and what is extraneous to this determination. Experts at the CTC must analyze the network data captured by the contractor and correctly associate the individual records associated with each shared data attribute. This expert intuition is used to both refine the profile for the data sharing agreement by expressing subcategories of data which need or need not be shared, which introduces the ability to perform redactions or perturbations of these data. Experts will also have an intuition for what data types should not be recombined, expressed as prohibitions for certain data append actions. For example, as seen in (2), employee identification numbers are unlikely to be necessary in determining the type of attack on a network, yet this data is likely to be present. Experts may express that this data is redacted in the initial profile, with the knowledge that this will not have a detrimental effect on the data utility.

To subsequently identify the threat, a query must be used to match the given threat information with the likely threat. Since there will be a collection of possible threats that match the threat information shared with the CTC, the probability of ascertaining the correct threat is given as follows:

$$P(correct\ identification \mid threat\ identified) = \frac{1}{count(identification\ query\ results)} \tag{4}$$

From (4), given that the threat has been identified, the probability of the correct threat is dependent on the number of results given by the identification query. As evidenced by the categorical re-identification approach proposed by (Sweeney, 2002), the number of possible results (threats) is proportional to the uniqueness of the threat with respect to its other categorical attributes. For example, a threat that is known only to be employed by one organization is much easier to identify in comparison with a threat that is employed by many organizations. A threat associated with one individual is far easier to identify compared to one employed by thousands. This determination can be further bolstered by expert judgement. The expert may perform additional queries on the resultant data in order to narrow down the threat. This process is based on heuristics they have developed through experience with specific types of threats. Given the knowledge, or (based on expert analysis) sufficient probability of identifying the threat, the CTC may now issue the appropriate mitigation response.

THIS PAGE LEFT INTENTIONALLY BLANK

Conclusions and Lessons Learned

As a result of the design of this experiment, and the implementation of the process to generate synthetic data, there were several conclusions that led to some important lessons learned. In this section, we detail the lessons learned and the technical challenges introduced by these findings.

Designing the evaluation functions to determine data utility is extremely difficult, because they are strictly bound to the queries on the dataset that are being executed. Each query is coupled with the business value derived from this data. Without knowing ahead of time what queries are necessary to derive the utility from the data, there is no available concept which allows an analyst to determine the confidentiality impact. This is important because without such knowledge, intuition is given about the requirements of the confidentiality preserving data retention actions that must be employed. While there are finite combinations for data append with respect to data attributes specified for sharing between any two parties in a data sharing agreement in general, it is impossible to determine the impact of prohibiting appending, or requiring redaction and/or perturbation of data, without knowing how the data will be used. This includes explicitly specifying what data will be appended together. Implementers must recognize that the queries they use have confidentiality risks built in.

In the example of the CTC and the contractor sharing threat intelligence, an analyst may wish to run a query appending commercial data; it is critical then to calculate the confidentiality impact that results from this query.

We have proposed a method to calculate and therefore engineer the confidentiality impact, but implementers, system designers, and acquisition analysts who are integrating systems must have an intuition that the application they are working on may have an inherent confidentiality risk. Without this intuition, and subsequent analysis, the confidentiality risks in the system will be increasingly obscured as attention shifts toward extracting more utility from data, rather than minimizing the confidentiality threat.

THIS PAGE LEFT INTENTIONALLY BLANK

Future Work

Based on the lessons learned from our current stage of research and development, we envision two fruitful avenues for future work which would advance the state of the art.

The first main area of future work would be to decrease the reliance on expert analysts and human decision making. This would be possible through the use of machine learning approaches, which would be capable of mimicking the same reasoning process that experts would use to make judgments about segmentation and separation of the data. In the sharing agreements from our running example, a machine learning algorithm could analyze the packet streams and perform classification of the data which would elicit the data attributes that are present in the data. Then, further processing could separate the data, record by record. The resulting separation heuristics would form the basis for the rules which would be present in the Eddy language – the machine learning algorithms would help to tailor the data retention specification to the specific data present in each sharing instance. The Eddy tool would then be used to validate the resultant specification, making guarantees that it is free of conflicts. As a result of this approach, the combined knowledge of many experts would be used to automate portions of the tool-assisted process which require significant time for experts to iterate over. The machine learning approaches could also be used to assist experts, rather than acting as a replacement for their judgement in that portion of the analysis.

One major limitation of the current Eddy tool is that it does not provide any ability to enforce the data retention specification. Rather, Eddy provides a requirements framework with built-in guarantees of confidentiality and consistency for which an architecture would be developed that performs the actual data actions. There exist novel mechanisms (Birrel & Schneider, 2014; Pearson & Mont, 2011)that can perturb or accompany data in ways that have predictable results if the data is used in a way that is specified by the Eddy profile prohibits. This could even be included as an additional data retention action. For example, if the data is shared with the intent of never being used for a specific purpose, it could be seeded with data such that if used for that purpose there would be an obvious downstream effect. Provided feedback mechanisms are present that allow the data to be recollected by the upstream party, additional assurances and compliance checks could be made such that the data in Eddy profiles is only used as specified. These mechanisms could be incorporated into, or complementary to, architectural mechanisms that assure compliance with the specified data practices.

Works Cited

Baader, F., Calvenese, D., & McGuiness, D. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications.* London: Cambridge University Press.

Birrel, E., & Schneider, F. (2014). *Fine-Grained User Privacy from Avenance Tags.* Ithaca: Cornell University.

Breaux, T. D., Hibshi, H., & Rao, A. (2014). Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requirements Engineering, 19*(3), 281-307.

Breaux, T. D., Smullen, D., & Hibshi, H. (2015). Detecting repurposing and over-collection in multi-party privacy requirements specifications. *Requirements Engineering*.

Danise, A. (2016, January 23). Will driverless cars mean the end of auto insurance? *Christian Science Monitor*.

Defense Science Board. (2013). *Task Force Report: Resilient Military Systems and the Advanced Cyber Threat.* Washington DC: United States Department of Defense.

Diez, O., & Silva, A. (2013, January). Govcloud: Using Cloud Computing in Public Organizations. *IEEE Technology and Society Magazine*, 66-72.

Duhigg, C. (2012, February 16). How companies learn your secrets. *New York Times*.

Dwork, C. (2006). Differential privacy. *International Conference on Automata, Languages, and Programming*, 1-12.

Federal Bureau of Investigation. (2014). *SASC investigation finds Chinese intrusions...* Washington DC: United States Senate Committee on Armed Services.

Garcia, M., Lefkobitz, N., & Lightman, S. (2015). *Privacy Risk Management for Federal Information Systems.* Gaithersburg: National Institute for Standards and Technology.

Horty, J. F. (1993). Deontic logic as founded on nonmonotonic logic. *Annals of Mathematics and Artificial Intelligence*, 69-91.

Lovlace, R. (2016). *Spatial Microsimulation in R.* Online: CRC Press.

Mooney, C. (1997). *Monte Carlo Simulation.* Sage Publications.

Pearson, S., & Mont, M. (2011). Sticky Policies: An Approach for Managing Privacy Across Multiple Parties. *IEEE Computer*, 60-68.

Ross, R. (2012). *Evolving Cybersecurity Strategies: NIST Special Publication 800-53.* Gaithersburg: National Institute of Standards and Technology.

Stamber, K., Brown, T., Pless, D., & Berscheid, A. (2013). Modeling and Simulation for Homeland Security. *20th International Congress on Modelling and Simulation.* Adelaide.

Sweeney, L. (2002). k-Anonymity: a model for protecting privacy. *International Journal for Uncertainty in Fuzzy Knowledge Based Systems*, 557-567.

United States Department of Veterans Affairs. (2014). *VistA 4 Product Roadmap.* Washington DC: Department of Veterans Affairs.

Vascik, P., Ross, A., & Rhodes, D. (2015). A Method for Exploring Program and Portfolio Affordability Tradeoffs Under Uncertainty Using Epoc-Era Analysis: A Case Application to Carrier Strike Group Design. *Naval Postgraduate School Acquisition Research Conference.* Monterey: United States Naval Postgraduate School.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net