Big Data, Little Security:
Addressing Security Issues in Your Platform

Thomas Macklin, Naval Research Laboratory\textsuperscript{1}
Joseph Mathews, Naval Research Laboratory\textsuperscript{2}

Abstract- This paper describes some patterns for information security problems that consistently emerge among traditional enterprise networks and applications, both with respect to cyber threats and data sensitivity. We draw upon cases from qualitative studies and interviews of system developers, network operators, and certifiers of military applications. Specifically, the problems discussed involve sensitivity of data aggregates, training efficacy, and security decision support in the human machine interface. While proven techniques can address many enterprise security challenges, we provide additional recommendations on how to further improve overall security posture, and suggest additional research thrusts to address areas where known gaps remain.

Introduction

Big data is a topic of much active research and development. While the precise meaning of the term can be debated, it is generally characterized as a collection of large data where the volume, velocity, and variety requires optimized approaches to process it \cite{1}. Regarding information security in this context, many organizations implement and rely on a variety of tools also characterized by big data. In some instances, these tools may hinder mission execution in situations where fewer alternative security controls could prove more effective. While a comprehensive review of such cases would be beneficial, this paper will focus on a few representative cases.

The title of this paper may be misleading, as it is not meant to critique any particular lack of security. Rather, this paper examines instances where ubiquitous (i.e., big) security controls actually undermine security and where targeted (i.e., little) security controls enhance the overall security posture more effectively. The latter is especially true of the Human Computer Interface (HCI), the mechanism by which users interact with computers. Just as many datum make up big data, many little security mechanisms can enhance the robustness and resiliency of an information-processing ecosystem. This paper discusses several scenarios where small, targeted security decisions have significant impact. The intent of this paper is to expose stakeholders outside the information security community to actions they can take to improve their organization’s security posture in nontraditional ways. Given that many people learn best about computer security through stories and anecdotes \cite{2}, this paper analyzes two specific cases and two recent security vulnerabilities. For each instance, we focus on three particularly relevant aspects:

1. “Data Aggregation:” Protecting sensitive information in the context of big data.
2. “Training:” Informing individuals to help them correct security decisions.
3. “HCI:” Empowering analysts with tools that help them make better decisions.

\textsuperscript{1} Thomas.macklin@nrl.navy.mil
\textsuperscript{2} Joseph.mathews@nrl.navy.mil
Security is rarely an end itself, but rather a function that organizations employ in support of a higher objective such as profit, new knowledge, or the production of goods and services. In each case, an organization’s most valuable assets include its raw data, the processes employed to turn that data into information, and the information products it produces. Thus, an organization’s sustainability ties to its ability to protect its assets from theft, corruption, or destruction. This paper provides researchers, developers, engineers, and information security principles with ideas on how to improve their overall security posture.

Case Study: Secure, Distributed Coalition Information Sharing

The first scenario examines the implementation of a multilevel web portal supporting distributed information sharing among coalition users [3]. The system was subject to rigid security requirements for network separation, network integrity, and data confidentiality. The system’s target users also had mission objectives that required them to share information at varying levels of sensitivity amongst team members, based on the sensitivity of the information ("need to know") and the relevance of the information to various groups of users ("need to share"). Users had access to a variety of familiar tools such as XMPP chat, Microsoft SharePoint, and email. These traditional tools were instantiated among disparate networks, each with its own security policy. The multilevel web portal, however, was the only tool spanning all networks and implemented security controls among all components (e.g., OS, database backend, HCI) to ensure each network’s security policies were properly enforced. Users evaluated the multilevel web portal alongside other available tools and surveyed on the experience. The primary objective of the experiment was to determine whether users would adopt a tool with more security controls if it allowed them to work from a single system, as opposed to using separate systems. After the experiment, users and several key participants responded to the following question: “Do you think the additional security protections provided by this tool merit switching from the existing tools?” with the following representative answers:

1. “No. The application was difficult to use, and security is the responsibility of the network. All of our workstations are compliant with the security policy, so that’s enough [security].”
2. “I don’t think so. The network is secured by a bunch of security tools, so our applications are secured too.”
3. “Our users know how to use [Microsoft] SharePoint, so they won’t be able to do their job unless the tool works like that. If you could make it work more like SharePoint, then I would say yes, but since it doesn’t then no.”
4. “Yes. I don’t really like the tool as much as what we have currently. If it allows me to do my job from my primary network with confidence that I’m secure, then I think we should learn to work with it and use it anyway.”
5. “I don’t think we should switch. If our existing tool were taken down because of a security failure, we’d just find another tool that we would use. Information is like water: when it wants to flow it will find a path.”

These answers illustrate the tension between the users’ mission objectives and the system’s security requirements. In cases where the users answered the question affirmatively, their comments suggested a high valuation of the security mechanisms implemented by the system. In cases where the answers were negative, their comments suggested an understanding of security that differed from the organization’s view, or a belief that the organization’s security controls (e.g., network separation,
authentication, access control, firewalls) were already sufficient. The multilevel web portal was designed to facilitate collaboration while minimizing the risk of confidentiality or integrity failure due to a network security breach or data spillage. Regardless, users had their own mission objectives that did not explicitly involve these security requirements. Survey results indicate the majority of users focused on accomplishing their objective as efficiently as possible by using whatever tools were available, and that they saw connectivity as a greater driver (i.e., enhancing participation) than protecting networks or maintaining confidentiality. This dichotomy illustrates a key challenge in achieving robust information security: reconciling the security objectives associated with executing a mission with the security objectives of those responsible for protecting networks and information.

**Data Aggregation**

The security objectives of the organization included network separation and data confidentiality, while the security objective of the users tied to availability. In order to meet their objectives, users tended to find the easiest place to share data at any given time. In cases where one tool was unavailable, users switched to the next best tool available. The aggregate of the objectives led to a paradox: in order for the users to meet their mission objectives, they actively worked against the security objectives of the organization. Both parties sought resiliency, albeit in different forms. While the multilevel web portal met the security and efficiency needs of the organization, it did not meet the operational needs of its users. As a result, users adopted the tool only when it was the best means by which they could achieve their objective. This occurred when the organization made the tool the only means available.

**Training**

All users received both Information Assurance (IA) training and system-specific training prior to the experiment. The IA training focused on rules which users were expected to adhere to during the experiment, while the system training focused on how to use the multilevel web portal. The experiment and surveys indicate both trainings adequately satisfied their objectives. The failure then was not in the training provided, but in the training not provided. Could training have helped improve acceptance of the new tool? Perhaps more rigor should have been applied to communicating how the portal helped the organization achieve its objectives, and how the additional controls (i.e., data confidentiality and network integrity) would also help them be successful. Experiments such as this and [4] [5] indicate that users often assume their systems prohibit bad security decisions, and that users will often make suboptimal security decisions if prompted with a choice. Furthermore, the post-experiment interviews indicated that users did not even understand how data aggregation could pose a security problem with regard to confidentiality. If the training covered security concepts and objectives in addition to specific compliance rules, users may have been less inclined to put all data in the common store most available at the time. As this type of real-world data can be garnered through field studies, we recommend more experiments that better capture and understand the dynamics of evolving security applications and environments.

**HCI**

When provided a choice between tools that made security decisions more intuitive and tools that made mission execution more intuitive, users overwhelmingly favored the latter. Of the over fifty users surveyed, only two stated that they preferred the multilevel web portal because it facilitated better security decisions. Furthermore, none of the other preferred tools met the same security requirements, and required installation, configuration, and management across separate networks. The data from this
and other studies such as [5] suggest that users will strongly prefer tools that fit into their mission workflow in a manner that they understand to support their objectives. Perhaps the most important lesson learned from this experiment is that security features are embraced only when users understand their benefits and they add efficacy to their workflows.

Field Study: Security Override Feature

Information systems typically implement security controls in a manner that users cannot override. In one instance of a security assessment for an integrated data analysis platform, we discovered that overrides existed for almost every security feature present in the system that prevented users from sharing information products. In each case, the override prompted the user according to the following scheme:

1. Inform the user of what action they were attempting to take.
2. Inform the user of the security feature they would override.
3. Inform the user of the consequence and/or outcome of this decision.
4. Ask the users whether they wanted to perform the action.

While this scheme was clear enough, it seemed rather unorthodox to trust users with security decisions of such significant consequence. Given the existence of such a prompt arguably does not constitute a high assurance system, further study was conducted where we discovered the following additional controls in place:

1. All users were thoroughly trained on the system’s security objectives and requirements.
2. All users were aware of the impact of a security failure on both their mission and organization.
3. All data was aggregated into a common repository, regardless of sensitivity, to facilitate analysis.
4. The override features were carefully designed and documented, and had been approved within all levels of the organization.
5. All override events were audited.

Information security professionals may question whether such an approach is optimal with respect to the system’s overall security posture. An analysis of this approach is provided.

Data Aggregation

Data aggregation was the core security problem behind the need for an override feature. The system was designed to store all data in a common location in order to maximize the users’ effectiveness in mining that data for useful information. However, this decision introduced an unforeseen quandary: once high value information was synthesized from the data and identified for dissemination, users lacked a reliable mechanism to assess whether sharing a given information product would result in a security failure. If users were given the ability to share all information products, they might cause a security failure by releasing sensitive data. However, in a case where users needed to release a questionable product, a preventative security control could cause an unacceptable mission failure. Users required a mechanism by which they could release whatever information products the circumstances required, while making good, traceable security decisions. Furthermore, the users themselves were information aggregators; they had extensive knowledge of the means by which they (and their tools) annealed data into information. While the organization could control all data and information products in the system, users harvested a mix of data with varying levels of sensitivity through mental models
resistant to technical access control mechanisms. This required the security enforcement mechanism to be user-driven. However, with better data separation and/or tools for measuring aggregated sensitivity, these controls could have been implemented in a manner less reliant on user discretion. A reasonable policy towards addressing the challenge of aggregation sensitivity is provided in [6], and discussion on future work to these ends is provided in the conclusion of this paper.

Training
The system aggregated data with a wide range of sensitivity levels in order to maximize the effectiveness of its users, but lacked a reliable means of determining the exact sensitivity of finished information products. Thus, release decisions were highly subjective with respect to confidentiality. If data aggregation poses a substantial risk of data leakage in this scenario, training proves a strong tool in mitigating that risk. As is typically the case, users were trained on rules they were expected to comply with and the intended operation of the tools, both with respect to security and otherwise. Users also received extensive training on the organization’s security problems and objectives, how this system adds value to the organization with respect to its mission effectiveness, and the possible adverse impacts of various security failures. While technical controls to manage the sensitivity of aggregated data and information products are preferable, the organization posited in its security plan that holistic training was core to its security plan and provable via operational metrics.

HCI
While the organization was unaware of field studies such as [3] [4] [5], their implementation kept with the principles learned in each. Users were provided with clear, user-centric explanations of the consequences of their security decisions, and each decision tied a security objective to a mission objective. Each control was also tied to training about not only the “what’s” and the “how’s”, but also the “why’s” of each decision point. This implementation demonstrates the notion of organizational risk management as intended by policies such as [7].

Vulnerability Exemplars- McAfee Antivirus, FireEye NX
Information security solutions use a variety of software and hardware to provide one or more security controls. Like any other system, these products can contain defects that can expose vulnerabilities or be exploited. While there are many examples of how security products can degrade security, this paper will focus on two relevant exemplars: the McAfee Antivirus for Linux defect described in CERT Vulnerability Note VU#245327 [8] and Vulnerability 666, a defect in FireEye appliances discovered by Google’s Project Zero team [9]. The discussion in this paper focuses on the less obvious implications of these vulnerabilities as they relate to the field of information collection and analysis.

McAfee Antivirus for Linux
McAfee Enterprise for Linux is an endpoint antivirus product with robust management and reporting capabilities [10]. The software detects security failures (e.g., introduction of a virus or malware) and ensures that the event is isolated and reported. The vulnerabilities described in [8] are exploitable in a manner that allows remote code execution with root privileges. These elevated privileges allow an attacker to gain access to a variety of resources including the logs collected by the antivirus software itself.
FireEye NX 7500 Appliance
The FireEye NX 7500 platform is a network security appliance meant to provide organizations a means to detect and prevent a variety of security failures [11]. The FireEye platform performs extensive traffic monitoring and analysis and keeps detailed logs regarding network activity. The vulnerabilities described in [9] allow an attacker remote execution of arbitrary code within the FireEye appliance itself. This level of access would allow an attacker to access the system’s logs or monitor traffic with the rights of the affected appliance itself.

Data Aggregation
Both products gather detailed data regarding usage patterns and report that information for analysis. The information is meant to improve security by informing operators about security events and actions that could be taken to address these events; ideally correlating users to the events so that the operators can identify affected assets. In addition to helping implement a useful security control (i.e., continuous monitoring), aggregating log data also poses a risk to the organization. If an adversary were to access to this log data, they could mine it themselves. Such analysis would yield information that could help adversaries identify the objectives, methodologies, and usage patterns of both the organization and the users of its networks. While other protections may mitigate the risks associated with the vulnerability (e.g., IPS, firewalls), the aggregation of such sensitive information poses a greater risk, where sources and methodologies are often among the organization’s most valuable assets.

Training
It is difficult to determine how training could help mitigate the risks associated with increasing sensitivity of the security infrastructure’s audit trail. Perhaps training administrators and developers about the risks that such aggregations pose to an organization may cause them to design, develop, implement, and maintain systems in a manner that would mitigate the risk.

HCI
As is the case with training, it is difficult to determine how HClis could be improved to address security-sensitive log aggregation. In some cases, involving the monitored subjects could be beneficial. As is described in [12], systems are frequently designed to assume that users will undermine built-in security mechanisms, and as a result minimize their insight into how a system is monitoring activity and/or enforcing a security policy. As is the case with user-facing privacy tools such as Lightbeam [13], interfaces that depict user activity in a graphical manner might make users more aware of potentially dangerous aggregations and help them to take remedial steps.

Recommendations
This paper briefly describes some common security problems that are particularly acute for organizations that analyze big data in order to obtain valuable information. The studies referenced in this paper illustrate challenges that are difficult to solve with traditional security solutions and compliance-based training approaches. In order to address these problems, security professionals and the user community need to work in an interdisciplinary manner to understand the problems that each face, the objectives each typically pursue, and the tools that each use to pursue their objectives. With a proper understanding of the organizations’ objectives, applications, and workflows, security researchers
and developers can tailor security policies to address their security objectives. Similarly, with a better understanding of the security controls required to achieve their objective, organizations will be ready to make smarter investments in the areas that will best achieve their objectives. Many of the resultant security controls will be relatively small and/or targeted to a particular application. However, if the processes used to develop these solutions are reusable and the cost to develop and deploy these controls are low, then the aggregate of these controls will result in a security plan that is designed to address the problems posed by the aggregation risk of big data. Some recommendations along these lines are provided below.

**Least Privilege for Enterprise Security Solutions**

While traditional security solutions implement controls that help achieve security and compliance objectives, these systems can create new problems. One such problem is the aggregation of security data that could indicate the objectives and/or methodologies that analysts use (e.g., web searches, database queries). With access to these logs, an attacker could surmise critical information regarding an organization’s operations and capabilities. Another risk is the ubiquitous presence of software agents with elevated privileges. These agents often have root access and can access traffic that would otherwise be encrypted in transit or at rest. A security failure in these agents would cause the system to be vulnerable to a more damaging attack than it would be without the agents present. More work needs to be done to ensure that software meant to enhance security cannot be used to compromise security. Better self-protection for security software could be greatly improved by limiting the privileges of software agents (e.g., not running as root) through well-understood techniques such as software fault isolation, binary diversity, privilege bracketing and privilege separation. When buyers demand enhanced security as a precondition for purchase, suppliers often respond favorably. For example, the Navy Marine Corps Intranet (NMCI) was the largest information technology purchase of its time. The Navy insisted that the infrastructure through which mobile devices would send and receive emails meet a minimum set of security requirements. Blackberry was the only vendor that implemented a corresponding security solution, and as a result became the mobile device selected for NMCI.

**Domain Specific Security Training**

More research into the efficacy of security training is needed. Many metrics for the efficacy of security training measure a student’s ability to correctly answer questions about security, but little evidence asserting that these questions could be tied to security outcomes was found. In any case, [2] indicates that users learn better about security from informal sources than from traditional training. The “Security Override” scenario described in this paper also suggests that users may learn to make better security decisions when the training relates to their day-to-day objectives as well as the organization’s overall objectives. While further research in these areas would be beneficial, organizations could easily enhance any existing security training they employ (which is likely to be largely compliance-oriented) with informal and/or domain specific training.

---

3 Privilege bracketing is when an application runs with lower privileges.

4 Privilege separation is when work requiring different levels of access is separated into different programs running with different privileges.
Harmonizing Security and Usability

Most security tools aim to protect the integrity of a system. Most analysts know the sensitive information that needs to be protected and don’t require elevated privileges to access it. As a result, they can leak information by any number of methods that are difficult to track such as covert channels or side-band mechanisms. Applications should be designed to make better security decisions at the working level, with less focus on attempting to restrict users from activities that often are required for them to achieve their objectives. Tools that would help users track the overall sensitivity of their information products based on its contents and sources would be useful. For example, if an application generated a report using queries against both a patient information database and a diagnosis history database with a user ID key, then the application could detect the potential aggregation risk and give the user further guidance on how to manage the document’s sensitivity. There exists much ongoing research into security and usability, such as the NIST Security-Usability Working Group. Much of this research has to do with the usability of specific security products or tools. Alternative approaches such as those suggested in [15] study security tools in the context of a user’s primary task. Such an approach would help organizations assess the extent of which the security controls in an HCI help users make better security decisions, while measuring any negative impacts to productivity or quality.

Modeling Risk Aggregation

The security community should have a formal model by which practitioners can calculate how data aggregation promotes risk. A corollary regulation from the financial sector, BCBS 239, serves as an example. It provides a means to assess the financial risk a bank is exposed to as it aggregates investments with varying ratios of value-to-risk. There was a remarkable amount of mathematical rigor put into the models for measuring aggregate risk, most notably in [16]. One of the reasons it is difficult to implement controls for data aggregation is there are few tools that assist in performing measurement, modeling and simulation. A less sophisticated approach to managing this risk is to store data in logically disparate stores (e.g., data collected from different physical sensors in a geospatial application) that require different authentication mechanisms. As users and their software agents use an increasing number of credentials, security controls could be implemented commensurate with the resultant information products.

Conclusion

Traditional enterprise security products implement security controls, yet are insufficient at holistically addressing the security challenges introduced by big data. In order to address the problems posed by data aggregation, organizations must find new ways to safeguard their critical tools, techniques, and procedures used to acquire, maintain and analyze data. Of particular concern is the organization’s ability to improve the robustness of its security infrastructure, as most commercially available security software has access to all real and derived data present on the network. Furthermore, analysts and other users of information technology need more effective security training that will help them understand the specific threats they face, how their security choices will impact outcomes, and how to reconcile security objectives with mission objectives. Finally, system designers need to consider the security implications affecting user-facing tools. Users would benefit from features giving them security-relevant feedback throughout their workflows, rather than using a separate security tool for forensic

validation. Measuring the impact of these "little" security mechanisms may prove challenging and problematic. However, field studies such as those cited in this paper suggest that they can have a large impact on scaling security in a manner that paces with the scale of data that they protect.

Bibliography


