Rividium Whites - White papers on leading edge technologies

Preventing Information Islands

Avoiding the Trap of Stand-alone Data

Andrew Bruce, CISSP, PMP, FITSP-D
CTO, RiVidium Corporation (http://www.rividium.com/)
andy.bruce@rividium.com

30 June 2010

Topic Summary:

- What are Information Islands?
- Brief history of computer infrastructure and current organization
- Formal security models and data management

Section 1: Introduction

Information management provides the key for effective decision-making. Whereas *data* can be thought of as individual facts (such as the number of people in a small town),[1] *information* is data organized to provide an answer to a complex problem (such as how much tax revenue can be expected to be gathered in a given year from an entire state).[2] A key problem in information management is that of *information islands*, a situation where data (such as financial data) is inaccessible from other necessary systems (such as regulatory compliance). These islands arise out of a *lack of trust*; individual information owners must be assured that all data is used correctly in order for sharing to occur. The Army and other large organizations see this problem as a major roadblock to integrate their information management capabilities ("We have a lot of innovation happening with the Soldiers, but it's happening in pockets and in silos, and in a way that's not shared with the rest of the Army").[3]

In this paper we look at *computer infrastructure* and *data management* and examine how they help us to build and keep the trust necessary to prevent information islands from occurring. In our organization, we manage data in both of its primary forms: centralized and decentralized. Centralized data management allows us to ensure that information is controlled and available regardless of the logged-in user's location, while decentralized data management allows us to provide modern Web services for sharing data with our clients and partners in a secure manner.

Section 2: Computer Infrastructure

Section 2.1: History

The key to information management is a solid and reliable computing infrastructure, with the systems making up that infrastructure constantly protected and maintained. In the early days of computer processing, this protection and maintenance was more straightforward than it is today. Popular literature from the 1950s depicted a world where computers were separate from everyday life, one connected to a mainframe server only in a very controlled environment.[4] Where multiple computers were connected, the connecting network cables were carefully protected from emanations and run through locked and hardened containers.[5] The machines themselves were huge and complex--consider the ENIAC project begun in the middle of World War II to calculate artillery tables with its almost 19,000 vacuum tubes,[6] the UNIVAC of the 1950s with its massive block appearance,[7] and the 1960s IBM Model 9020 (consisting of *three* System/360 Model 50s, designed to continue running in real-time even if two of the three systems failed).[8]The biggest risks to the machines and the information they represented were component failure (such as vacuum tubes) and human failure. Human failure resulted from the small group of people that really understood and could use computers, primarily from direct sabotage or outright theft.[9] Managing information in this environment meant keeping the data center secure and highly protected through such methods as two-man control to prevent unattended access, glass walls offering full visibility into the data center, and entry doors staffed by guards.

Computer infrastructure has always involved more than simple policing, however. The Parkerian Hexad[10] defines the key concepts of *availability* and *utility*; computer systems must be capable of providing useful information upon demand. Early computer systems such as ENIAC and UNIVAC had numerous and regularly-failing vacuum tubes (one at least every two days)[11] that required a constant supply of spare parts. System managers had to understand the limits of their computer systems and provide for rapid and efficient repair of these parts. Also, these early computer system designs constantly evolved; consider that the UNIVAC serial number 3 featured water cooling while serial number 1 used air cooling. The wise administrator understood, anticipated, and mitigated downtime ahead of time by understanding her environment intimately.

Section 2.2: Our Organization Today

We address infrastructure needs by ensuring that we have a reliable and predictable computing environment. We create this environment by: a) providing massive amounts of shared storage; b) using redundant power supplies to maximize system up-time; and, c) using virtual machines to centralize management.[12] Additionally, we make strong use of *attribute-based access control* (ABAC); basically, we tie information access not to hard-wired access control lists (ACLs) but to roles associated with the logged-in user. Although we are a small company, we are able to guard our data well; this level of assurance is our major tool for preventing the occurrence of information islands.

Section 3: Data Management

Section 3.1: History

All the way up through the 1970s, data was often stored as punched-cards, computer printouts, or magnetic tape.[13] "Losing"millions of data records (such as the Veterans Administration fiasco of June, 2006[14]) required substantial quantities of bulky media to be misplaced or stolen. Thus, information management concentrated on physical controls: techniques such as manned entry points to limit access, highly controlled physical communication links (prearranged and authorized between trusted points only), two-man controls, and dumb terminals all helped to keep data where it belonged: in the data center.[15] Data stored on this media was subject to classification (data classification applies both to commercial and military environments although it is primarily associated with the military).[16] In the military these classifications range from UNCLASSIFIED (non-sensitive data) to TOP SECRET (disclosure can cause grave damage to national security).[17]Strict controls were used to ensure that only those users who had authorization and need-to-know could access the critical data of computer programs.

Section 3.2: Data Management and Security Models

As computer information systems evolved, logical data access grew as an issue; system managers no longer worried solely about controlling access to punched-cards, magnetic tapes, and paper reports. Now they had to consider computer screens, ever smaller disks that could store ever larger amounts of data, a plethora of network connections that could access computer information, and "covert attacks" whereby malicious programs or users could act against data or try to access data inappropriately. To help ensure that systems (and their data) remained secure, a set of *security models* were developed. The goal of these security models was to express the security requirements for a given system precisely. If the security model was implemented correctly, the system manager could have confidence that the data the system contained would remain in a reliable state.[18]

Section 3.2.1: Bell-La Padula and Confidentiality

In 1975 the Bell-La Padula (BLP) formal security model was developed for the United States Air Force by the Mitre Corporation specifically to address confidentiality.[19] In this model, stored data elements ("objects") each have a classification while data consumers ("subjects") have a clearance. To control the access of subjects to objects, the model defines three basic data access properties:

- Simple Security Property: A subject cannot read an object with a higher classification than the subject's clearance ("no read-up").
- *-property ("star property"): A subject with a higher security clearance cannot write to an object with a lower classification ("no write-down;" for example, a TOP SECRET subject could not write a data file that is classified as SECRET).
- Strong *-property: A subject may write objects only where the security levels match.

Additionally, BLP provides the *tranquility model* which states that an object's security classification may not change as a result of operations. The net effect of this model is that for an object to change its classification (for example, from TOP SECRET to SECRET) the operation must be performed by a *trusted subject* (a subject explicitly granted permission to make this type of change).

Section 3.2.2: Biba, Clark-Wilson, and Integrity

Integrity relates to the wholeness and completeness of data[20] and provides decision makers with the reliable view of data necessary to function effectively. Integrity failures can be quite damaging; if a company's accounting data is modified incorrectly then all sorts of bad things can happen (such as malicious changes to personal and confidential data). The integrity security models exist to address this danger.

Section 3.2.2.1: Biba[21]

The Biba model is similar in form to Bell-LaPadula (and both are state-based) except that the rules are inverted and the goal is data integrity rather than data confidentiality.

- Simple Integrity Property: A subject can read an object only if the integrity level of that object is greater than or equal to the subject (no read-down). A practical way of looking at this is in terms of a research paper: one cannot quote from an arbitrary source from the Internet because that source is not highly trusted. Instead, one must quote from only known and trustworthy sources.
- *-integrity ("star integrity") property: A subject can only write to an object if the subject *dominates* the object (has an integrity level greater than or equal to the object, also called "no write-up"). In our organization, we run a Web log ("blog"): only blog administrators may approve the content of new posts. If users could create and approve blog posts without verification, then our Web site would no longer have the same level of integrity.
- Invocation property (execution integrity property): A subject cannot invoke subjects of higher integrity. This property ensures that lower-integrity subjects cannot attempt to work around the

simple integrity property by the expedient of executing a more trusted process. In our organization, we apply this model in our UNIX operating system with “secure binaries” such as **/usr/sbin/reboot** (reboots system). These are trusted programs that only system administrators can execute because the programs can change system data.

Section 3.2.2.2: Clark-Wilson[22]

The Clark-Wilson security model guards integrity by separating data into two categories: *unconstrained* (not highly protected) and *constrained* (highly protected). Users may modify unconstrained data items directly; this information includes such elements as a user's contact phone number or salutation. Constrained data items are a different matter; users may request modification only by way of a *well-formed transaction*, or WFT. WFTs require a user to initiate a data modification process via a *transformation procedure* (TP), which is a software application that carries out the operations on behalf of the user. The TP operates on the constrained data item (CDI), and the combination of these three elements (user, TP, and CDI) form an *access triple*. To ensure proper enforcement, a separate *integrity verification procedure* (IVP) must “bless” each CDI updates before it can be considered complete.

In our organization, we use this model extensively to enable secure decentralized information management. Consider a standard Web-based shopping cart such as we build for customers regularly: two online users are viewing merchandise from a major vendor (for example, AOpen motherboards) but through two different online resellers (let's say Newegg.com and CDW.com). Initially, each user sees the same information on her screen indicating that ten motherboards remain in stock. The first user puts in a request for eight motherboards and the second user puts in a request for nine. Both CDW and NewEgg serve as the transformation procedure because each is an intermediary between the customer (user) and the constrained data item (the order to AOpen). In this case, both CDW and NewEgg verify and approve each order based on the information they have. However, AOpen has the final say – it provides the integrity verification procedure to reconcile these mutually exclusive orders. The net result: the first user gets her eight motherboards while the second user gets only two.

Section 3.2.2.3: Goguen-Meseguer[23]

Goguen-Meseguer (GM), our final security model, provides a number of critical features to information security and also helps to work around some of the restrictions of Bell-LaPadula (BLP). Remember that in BLP a subject with a higher clearance could never write data to a lower classification (TOP SECRET to SECRET, for example) because inadvertent information disclosure could occur. Goguen-Meseguer addresses this by providing the concept of *noninterference*: the actions of one user do not impact what a user with lower classification can see.

In our organization, we have used this model to manage information for a secure system. While we can't provide specifics, we can walk through a notional example: an enemy agent posing as a shipping clerk may suspect that a given ship contains military components and thus wants to be aware when that ship weighs anchor (CLASSIFIED information). The clerk can't get to that information directly via the BLP simple security property (no read-up) , but the clerk *does* have permission to request that additional cargo be loaded onto the ship (UNCLASSIFIED request). The data leak can occur if the clerk attempts to modify the cargo and receives an error message. In that case, the clerk can infer that the ship has weighed anchor. By applying Goguen-Meseguer to this problem, modifications to classified information (the ship weighing anchor) cannot affect what the shipping clerk sees on the cargo change modification screen.

The above may seem to be contradictory; how can classified information be modified but still not visible to the lower-level user? The answer lies in the concept of *polyinstantiation*,[24] which states that a single object (such as the “still at anchor” indicator for our ship) can have multiple values depending on the classification of the user accessing the data. A change to this higher-classified setting is not visible to the lower-classified requester.

Section 3.3: Our Organization Today

We use the formal models above to manage our data (centralized and decentralized) as well as to ensure data confidentiality. BLP and Goguen-Meseguer are key components of a secure Linux operating system we maintain for a Federal customer; objects (programs, data, and so on) have classifications and categories associated with them and changes made by a privileged user may not impact an unprivileged user. We use the Biba model to ensure the integrity of our corporate blog, and we use Clark-Wilson to manage our Service-oriented Architecture[25] and online transactions. By taking these measures, we provide the *trust* necessary to prevent the growth of information islands.

Section 4: Information Management and the Future

Consider the present: USB sticks that can hold terabytes of information; wireless network gateways under constant; a computer-literate public expecting always-on and fully interoperable data; and numerous Federal regulatory requirements. But tomorrow offers even more challenges; with the advent of social networking and crowd-sourcing applications,[26] raw data replicates and mutates in ways that early data center managers could scarcely conceive. We can look forward to an explosion of data-centric applications using *semantic reasoners.*[27] These reasoners go far beyond simple pattern matching (such as a Google keyword search); rather, they look for embedded and implied relationships between data objects. This raises a whole new set of issues related to privacy and managing a rich online user experience as companies create increasingly vast data stores to track user habits and preferences. With this type of freewheeling and non-stop data evolution, we in Information Assurance can rest assured that our ability to manage information effectively will provide the critical backbone to support tomorrow's innovations.

Reference List and Endnotes

Al-kohafi, Majd, Su Chang, Thomas E. Daniels. "SCWIM an Integrity Model for SOA Networks" in *Proceedings of the 2008 IEEE International Conference on Web Services*. Washington, DC: IEEE Computer Society, 2008.

Asimov, Isaac. "Franchise." *If: Worlds of Science Fiction* (August, 1955). Quinn Publications.

Bell, David E., and Leonard La Padula. *Secure Computer System: Unified Exposition and Multics Interpretation*. ESD/AFSC, Hanscom AFB, Bedford, MA 01731, 1976 [ESD-TR-75-306, DTIC AD-A023588].

Bosworth, Seymour, M.E. Kabay, Eric Whyne, eds., *Computer Security Handbook: Volume 1, 4th ed.* Hoboken, NJ: John Wiley & Sons, Inc., 2009.

Department of Veterans Affairs . Office of Inspector General. *Review of Issues Related to the Loss of VA Information Involving the Identity of Millions of Veterans*. Washington, DC: Government Printing Office, 2006. [Report No. 06-02238-163].

Harris, Shon. *CISSP All-in-One (AIO), 4th ed.* New York: McGraw-Hill, 2007.

Hebeler, John, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez. *Semantic Web Programming*. Indianapolis, IN: Wiley Publishing Inc., 2009.

Hoover, J. Nicholas. "Gov 2.0: Army Announces Apps For Army Competition." *InformationWeek Government* (September 10, 2009), http://www.informationweek.com/news/government/enterprise-apps/showArticle.jhtml?articleID=219700596 (accessed June 21, 2010).

Kempf, Karl. "ENIAC." In *Electronic Computers within the Ordinance Corps*. Aberdeen Proving Ground: Aberdeen, MD, November, 1961. DTIC AD-0277129. Retrieved from http://ftp.arl.army.mil/~mike/comphist/61ordnance/chap2.html (accessed June 23, 2010).

Lohr, Steve. "The Crowd is Wise (When It's Focused)." New York Times, July 18, 2009, pg. BU4.

Merriam-Webster. *Merriam-Webster's Online Dictionary*. http://www.merriam-webster.com/netdict/data (accessed June 23, 2010).

Parker, Donn B. "A New Framework for Information Security."*Fighting Computer Crime* (New York: John Wiley & Sons, 1998).

Stephenson, Peter R., ed. *Information Security Essentials: Section 1*. Auerbach Publishing, ISBN 978-1-4398-0030-0, 2009. Retrieved on June 6, 2010 from https://norwich.angellearning.com/AngelUploads/Content/MSIA_2_0/_assoc/msia_s1/msia_s1_readings/ise_section_1_et.pdf.

[1] Merriam-Webster, "data," *Merriam-Webster's Online Dictionary*, http://www.merriam-webster.com/netdict/data (June 23, 2010).

[2] Merriam-Webster, "information," http://www.merriam-webster.com/dictionary/information (June 23, 2010).

[3] J. Nicholas Hoover, "Gov 2.0: Army Announces Apps For Army Competition," *InformationWeek Government*, September 10, 2009. http://www.informationweek.com/news/government/enterprise-apps/showArticle.jhtml?articleID=219700596 (June 21, 2010). In this article, Army Chief Information Officer (CIO) LTG Jeffrey Sorenson discusses the need for the Army to become innovative by allowing data to be shared outside of information islands ("silos").

[4] Isaac Asimov, "Franchise," *If: Worlds of Science Fiction*, August, 1955 (Quinn Publications). This fascinating short story presents a future where only one man has a vote. The master computer named "Multivac" simply asks a few questions and determines (with the help and interpretation of numerous computer "priests") what the results would be. In this worldview humans and computers are kept strictly separate; beyond robotic servants the concept of personal computing was not even imagined.

[5] Peter R. Stephenson, ed. *Information Security Essentials: Section 1* (Auerbach Publishing, ISBN 978-1-4398-0030-0, 2009), pg. 6. Retrieved on June 6, 2010 from https://norwich.angellearning.com/AngelUploads/Content/MSIA_2_0/_assoc/msia_s1/msia_s1_readings/ise_section_1_et.pdf.

[6] Karl Kempf, "ENIAC," in *Electronic Computers within the Ordinance Corps* (Aberdeen Proving Ground:, Aberdeen, MD, Nov 1961), DTIC AD-0277129. Retrieved from http://ftp.arl.army.mil/~mike/comphist/61ordnance/chap2.html (accessed June 23, 2010). The Electronic Numerical Integrator And Computer (ENIAC) was the world's first general-purpose electronic computer. Although roundly denounced as a "monstrosity" by some (such as Dr. Jay Forrester of MIT), the ENIAC was reliable and did its work well until it was immobilized by lightning in 1955.

[7] Seymour Bosworth, M.E. Kabay, Eric Whyne, eds., "Chapter 1.2.2: Large-Scale Computers," *Computer Security Handbook: Volume 1*, 4th ed. (Hoboken, NJ: John Wiley & Sons, Inc., 2009), pg. 46. Released in 1951 the "Universal Automatic Computer" (UNIVAC) contained 5,200 vacuum tubes and measured 25 feet by 50 feet.

[8] Paul E. Ceruzzi, *History of Modern Computing*, 2nd ed. (Boston: MIT Press, 2003), pg. 150.

[9] Bosworth et. al., *CSH*, Chapter 2.3.1, pg. 65. Also mentioned in the text is the sobering statistic that 25% of all computers have been physically assaulted by their owners.

[10] Donn B. Parker, "A New Framework for Information Security,"*Fighting Computer Crime* (New York: John Wiley & Sons, 1998), pp. 229-255. (Note: This chapter is included as Chapter 3 of the *Computer Security Handbook* cited above.) Mr. Parker critiques the classical *security triad* of Confidentiality, Integrity, and Availability and proposes a new taxonomy in terms of loss scenarios: Availability (timely access to resources); Utility (resources can be used, consider encrypted data where the key is lost); Integrity (resources are whole and consistent); Authenticity (resources are authoritative and trustworthy); Confidentiality (resource access is limited to those with need-to-know); and Possession (resources remain in possession of authorized owners).

[11] Ceruzzi, *History*, pg. 28. Vacuum tube failures, although common, were by no means showstoppers and could generally be corrected within fifteen minutes.

[12] Interview with Chief Network Engineer, June 24, 2010.

[13] Ceruzzi, *History*, pg. 69. In 1956, IBM released a multi-disk storage device that could contain five megabytes of data. Over the next decade and a half, disk storage improved and was commonly used for program storage, but data was primarily stored on magnetic tape.

[14] Office of Inspector General, *Review of Issues Related to the Loss of VA Information Involving the Identity of Millions of Veterans* (Washington, DC: Department of Veterans Affairs, July 11, 2006) [Report No. 06-02238-163], Introduction, pg. i.

[15] *ISE*, pp. 5-6.

[16] Shon Harris, "Information Security and Risk Management," *CISSP All-in-One (AIO)*, 4th ed., (New York: McGraw-Hill, 2007), pp. 117-118.

[17] Ibid, pg. 118.

[18] *ISE*, pg. 94.

[19] David E. Bell and Leonard La Padula, *Secure Computer System: Unified Exposition and Multics Interpretation* (ESD/AFSC, Hanscom AFB, Bedford, MA 01731 1976) [ESD-TR-75-306, DTIC AD-A023588]. This entire section is greatly paraphrased from this seminal work, which defines the eponymous Bell-La Padula (BLP) security model. The BLP is the most widely recognized model in existence.

[20] *CSH*, pg. 101, which contains an excellent definition of *integrity*. In the CSH, Chapter 3 ("Toward a New Framework for Information Security") is a direct reprint from Mr. Donn B. Parker's book *Fighting Computer Crime* cited above.

[21] *ISE*, pp. 96-98 provides brief definitions of the processes. Also useful was Shon Harris' *AIO*, pp. 336-338.

[22] Bosworth et. al., *CSH*, "Chapter 9.4.3: Clark-Wilson Model," pp. 290-292. Clark-Wilson is structurally similar to many modern computer processing elements; for example, Service-oriented Architectures closely mimic the processing of Transformation Procedures via the use of Web services.

[23] *ISE*, pp. 104-105.

[24] Harris, *AIO*, Chapter 11, pg. 930.

[25] Majd Al-kohafi, Su Chang, Thomas E. Daniels, "SCWIM an Integrity Model for SOA Networks," in *Proceedings of the 2008 IEEE International Conference on Web Services*, (Washington, DC: IEEE Computer Society, 2008), 675-682. In this paper, the authors discuss a modification to the Clark-Wilson integrity model named "Service Clark-Wilson Integrity Model (SCWIM)." The emphasis is on creating well-formed transactions in a distributed model with rollback (transaction consistency) support.

[26] Steve Lohr, "The Crowd is Wise (When It's Focused)," *New York Times*, July 18, 2009, pg. BU4. Crowd-sourcing provides the companies the opportunity to outsource tasks to the Internet audience as a whole. Consider Amazon.com's Mechanical Turk project (https://www.mturk.com/mturk/welcome) for an example of how numerous applications' data entry requirements have been outsourced to the Crowd.

[27] John Hebeler, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez, *Semantic Web Programming*, (Indianapolis, IN: Wiley Publishing Inc., 2009), pp. 4-9. This highly informative book leads the reader through a complete exploration of what an *ontology* means to current keyword-based searching and how this model will change drastically over time. The authors maintain an active Web site at http://semwebprogramming.org/ with blog entries, external links, and opensource projects. The author of this paper works actively with these technologies, specifically on an opensource project in the C# programming language to extend reasoning within the Department of Defense (see http://razor.occams.info/code/semweb/). The impact of semantic-based searches on the future of data assurance cannot be underestimated; once any data is scanned or stored in any location (for example, within search caching locations like Google), it is subject to being associated with individual users, communities of interest (COIs), and even specific categories of search queries. Privacy and data classification issues will be a huge concern as Web 2.0 (interactive Web) gives way to Web 3.0 (semantic Web).