Applying Security to an Enterprise using the Zachman Framework

Lori L. DeLooze

An enterprise information architecture provides a framework for reducing information system complexity and enabling enterprise information sharing. Much like a homeowner designing a home, information technology managers work with an architect to provide an agreed upon architectural drawing for the information and processes in the enterprise. This high level architectural drawing does not change with tactical decisions to deploy improved technology since it is simply built around a framework of business processes and the information that they need. Since most enterprises have existing information systems, the architectural drawing provides the future state and facilitates the best possible strategy to remodel with the least amount of inconvenience to the business.[1]

In today's environment, each department or system usually has a vertically integrated approach to data, process, and technology. For example, department A has an application with its own database and runs on its own computer. Department B has another application with its own database and runs on its own computer. The same is true for department C. The Zachman Framework, named after John Zachman, has emerged as a way to develop an enterprise-wide information architecture. This framework moves from this vertical, departmental approach to a completely opposite horizontal approach. Instead of representing the data, process and technologies as entirely separate entities; he organized them around the points of view taken by various players.[2]

The Zachman Model is actually a matrix with six rows and six columns. The levels of the matrix are organized around different points of view. Let's look at the framework using a building construction analogy. At the top level, the Ballpark View, the project is defined by a family's or a business' needs and zoning regulations. The construction of the project will be very different depending on whether it will be a single-family residence or a major shopping center. The next layer, the owner's view, looks at the desires of the users of the property in general terms, does the owner want two bedrooms and two baths or seven bedrooms and six baths. Does he want a basement? The architect's view then takes these considerations into account while designing a structure that not only meets the owner's requirements, but also satisfies constraints imposed by laws, regulations and industry best practices. The designer takes these architectural specifications, examines the individual components, and makes suggestions to make the living or working environment much more comfortable. At this stage, for example, a designer may suggest making a closet smaller so that a bathroom can be larger. Based on the architectural drawings and the designer's suggestions, the builder creates a blueprint and produces the finished product.

An upper row or perspective does not necessarily have a more comprehensive understanding of the whole than a lower perspective. Nor does an upper row decompose into greater detail in a lower row. Each row represents a distinct, unique perspective; however, the deliverables from each perspective must provide sufficient detail to define the solution at the level of perspective and must translate to the next lower row explicitly. Each perspective must take into account the requirements of the other perspectives and the restraint those perspectives impose. The constraints of each perspective are additive.

For example, the constraints of higher rows affect the rows below. The constraints of lower rows can, but do not necessarily affect the higher rows. Understanding the requirements and constraints necessitates communication of knowledge and understanding from perspective to perspective. The Framework points the vertical direction for that communication between perspectives.[3]

In addition to the rows down the left side that represent the various perspectives, the Zachman matrix includes the columns across the top that are the different focuses or product abstractions of these perspectives. Each focus asks a question. The way in which the questions are answered depends heavily upon the perspective. These are:

1. ***Data (what):*** Each of the rows in this column address understanding of and dealing with any enterprise’s data. This begins with a list of the things that concern any company in this industry, affecting its direction and purpose.
2. ***Function (how):*** The rows in the function column describe the process of translating the mission of the enterprise into successively more detailed definitions of its operations.
3. ***Network (where):*** This column is concerned with the geographical distribution of the enterprise’s activities. This ranges from a simple a listing of the places where the enterprise does business and how they communicate with each other to the specifications of the particular computers, protocols, and communications facilities at each location.
4. ***People (who):*** The fourth column describes who is involved in the business and in the introduction of new technology.
5. ***Time (when):*** The fifth column describes the effects of time on the enterprise. It is difficult to describe or address this column in isolation from the others, especially column two.
6. ***Motivation (why):*** As originally described, this column translates business goals and strategies into specific ends and means.

	Data (What)	**Function (How)**	**Network (Where)**	**People (Who)**	**Time (When)**	**Motivation (Why)**
Ballpark View	List of things important to the enterprise	List of processes the enterprise performs	List of locations where the enterprise operates	List of organizational units	List of business events / cycles	List of business goals / strategies
Owner’s View	Entity relationship diagram	Business process model	Logistics network (nodes and links)	Organization chart	Business master schedule	Business plan
Architect’s View	Data model	Essential Data flow diagram; application architecture	Distributed system architecture	Human interface architecture	Dependency diagram, entity life history	Business rule model

Designer's View	Data architecture	System design: pseudo-code	System architecture	User interface	"Control flow" diagram	Business rule design
Builder's View	Data design, physical storage design	Detailed Program Design	Network architecture	Screens, User Access	Timing definitions	Rule specification in program logic
	(Working systems)					
Function System	Converted data	Executable programs	Comms	Trained people	Business events	Enforced rules

Figure 1: The Zachman Framework

For security architecture modeling purposes, the first three columns of the Zachman matrix (data, function, and network) are extremely useful. They provide the answers to what data assets the organization controls, how they are used and where they are located. Similarly, the first five rows of the matrix give a unique perspective of a particular security challenge. The highest level, the Ballpark View, defines a clear and coordinated boundary (domain) of the system for the purposes of identifying the people, subsystems, and needs impacted by the system. The Owner's View captures the business and organizational relationships, and their external interfaces. It also documents sources of system requirements, including those derived from legacy systems. The Architect's View defines the functional capabilities of the system and establishes required interactions between subsystems. The Designer's View establishes and documents the security architectural design and provides a basis for system measurement. Finally, the Builder's View provides a detailed description of the design and methodology for monitoring and correcting system performance.

Each layer in the framework relates to a tool that can be used to secure the system. For example, an overall organizational security policy would be implemented in the Ballpark View. A tailored security policy is required that outlines the data to be handled, how the data is accessed, what controls are required on that data, and an indication of the organization's paranoia level. In addition, data needs to be grouped based on the impact to the business. Common groupings for data would be customer data (financial records, medical records, and order information), business data (company financial records, competitive analyses, and intellectual property), and employee data (salary, benefits, and personal information). The next step is to determine who needs access to these records and where they are located. Exposure points and threats will increase as you provide access to internal employees, external employees, business partners, customers, and third parties. The security policy must clearly outline levels of acceptable risk for each grouping of data. Risk analysis must consider the level of realistic threats to the system, the visibility of the organization, the consequences of an incident, and the organization's sensitivity to the intangible costs of an incident.[4]

The next level down, the Owner's View, considers the groupings of data and means of access available to both internal and external users to determine the placement of routers and firewalls as data control devices. Routers and firewalls control access so that only “approved” users and services are allowed. A packet filtering-router analyzes network traffic at the transport protocol layer. Each IP network packet is examined to see if it matches one of a set of rules defining what data flows are allowed. The rules determine whether communication is allowed based upon the information contained within the Internet and transport layer headers and the direction that the packet is headed. For example, to allow inbound and outbound HTTP service, the following rule set would be created.

Source	Destination	Source Port	Destination Port	Active Session?	Direction of Packet	Filter Action
Outside	Inside	80	>1023	*	Out	Allow
Inside	Outside	>1023	80	Yes	In	Allow
Inside	Outside	>1023	80	*	Out	Allow
Outside	Inside	80	>1023	Yes	In	Allow
*	*	*	*	*	*	Block

Figure 2: HTTP Router Rules

Similarly, a firewall can be inserted at a critical interface point to serve as an application-level or circuit-level gateway. An application level firewall evaluates network packets for valid data at the application layer before allowing a connection. The firewall examines the data in all network packets at the application layer and maintains the connection status and sequencing information. Specialized application software and proxy services are included in most application layer firewalls. Although application level firewalls provide increased security over a packet filtering router, they are much slower since inbound data is processed by the application and by its proxy.

A common router and firewall configuration is shown in Figure 3. This screened subnet has two packet-filtering routers and a firewall. This configuration creates an isolated subnetwork, which makes the internal network invisible to the Internet and the inside router advertises only the existence of the screened subnet to the internal network. There are now two layers of defense between the outside and the DMZ and three layers between the outside and the internal secure network. [5]

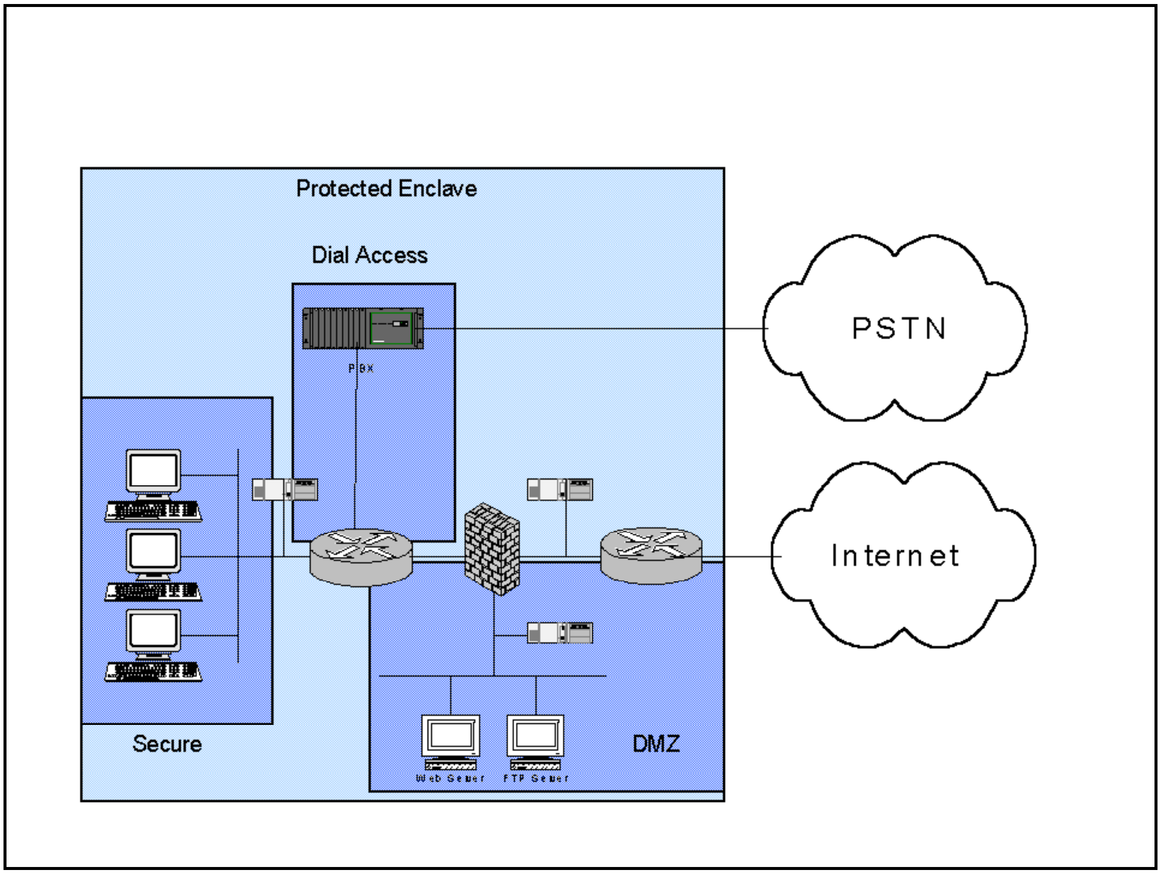

Figure 3: Sample Security Architecture

Because most of today's IT environments are so varied and complex, simply installing a firewall or other security measures can no longer guarantee long-term security. The next layer, the Architect's View, we will look at hardening the operating system and applications. A good hardening process eliminates most known vulnerabilities, giving the organization much more reliability and integrity of its services and information. Operating system hardening is the process of removing everything from the operating system except that which is absolutely necessary. A hardened system will disable unnecessary binaries (remote commands, sendmail, etc.), block all unnecessary ports and services (telnet, ftp, etc.), and be fully patched and tested. Application hardening means limiting the applications use to those aspects that are necessary for minimal user interaction. If you only use 25 of the 1000 features available, configure only the 25 necessary and omit the rest. These actions increase security over and above the level offered when installing the operating system and/or application out of the box. Any time the system administrators make a change or upgrade to the operating system and applications, configuration modifications should be considered to remove known vulnerabilities and common mis-configurations.[6]

At the next level, the Designer's View, we introduce mechanisms to protect the network and monitor system traffic. While the mechanisms employed at each of the above levels are used to prevent intruders, even the most secure systems are vulnerable to intruders and malicious activity. Intrusion detection systems and virus protection schemes are implemented at this level. There are two major types of IDSs; host and network-based. A network-based IDS monitors the whole network or a segment of the network, while host-based systems monitor a particular computer. A network-based IDS looks for security problems on the network, specifically analyzing the packets and looking for attack signatures. In contrast, a host-based IDS is installed on the computer or server that it is protecting, scanning the system logs for security warnings and other anomalies. [7] By using both these systems in conjunction, an intruder can be identified and ejected from the system before any damage is done or any data is compromised. Even if the detection is not timely enough to preempt the intruder, damage will be reduced and recovery will be quicker if the intrusion is detected sooner rather than later.

While an active intrusion can cause a great deal of damage, the introduction of a virus to a private network can be just as devastating. There are several ways that a virus can enter an organization, such as messaging, diskette sharing, FTP, web downloading, and hackers. The ideal solution to the threat of viruses is prevention. Although user education and limiting system services can reduce the number of successful viral attacks, complete prevention of viral infections is nearly impossible to achieve. Assuming prevention does fail, the viral infection must be detected, identified and removed. The most advanced antivirus techniques use heuristic rules to search fragments of code for structures associated with viruses and use memory-resident programs that identify a virus by its behavior rather than its form. These scanning and activity trap components limit the ability of viruses to penetrate a system and then limits the ability of a virus to update files in order to pass on the infection. [8]

Finally, the Builder's View identifies the access control mechanisms necessary to complement the security tools that were initiated in the above levels. At this level, strong passwords, Public Key Infrastructure (PKI) and Virtual Private Network (VPN) schemes are considered. PKI allows users of an inherently non-secure network medium, such as the Internet, to securely and privately exchange data by the use of a public and a private cryptographic key pair. The key pair is obtained and shared through a trusted authority. A VPN solution, in contrast, is a combination of hardware and software that will encrypt and encapsulate transmission packets of information, using a protocol tunnel through existing firewalls and security stacks. These encrypted and encapsulated packets will be recognizable only by a similar device that recognizes those packets.

These authentication techniques are used to determine whether information is trustworthy and genuine, that it has not been corrupted or fabricated. It includes mechanisms for determining whether actors – people and processes – are as they claim to be, and mechanisms for determining whether data have been tampered with or attributed to false sources. Because authentication is used as a means of controlling access to information and resources, it directly protects against many other unauthorized acts, including theft of sensitive information.[9] Installing the authentication mechanisms on the system is just like putting the locks on the doors and windows of a new home. This is the final step before turning a secure, fully functional system over to the proud new owners.

An enterprise information system architecture is made up of the information systems boundaries, the enterprise data hubs, and the data standards that integrate those information systems. Like any structure, an enterprise information system needs continuous maintenance and upkeep. You should expect to repaint and wallpaper occasionally, and over time, as the organization changes and grows, you may even need to remodel. Examining and testing new technology before actually incorporating it into the system architecture is much like getting a building permit for a new addition. After every major remodeling effort, however, the security architecture needs to be re-examined, using the Zachman framework as a guide. Designing and implementing a streamlined, integrated security architecture should not be difficult if you follow this stable, proven process.

[1] Cook, Melissa A., Building Enterprise Information Architectures, p. xx.

[2] Hay, David C**.,** THE ZACHMAN FRAMEWORK: AN INTRODUCTION, Essential Strategies, Inc.; www.tdan.com/i001fe01.htm

[3] Federal Enterprise Architecture Framework, Version 1.0; www.itpolicy.gsa.gov/mke/archplus/fedarchhtml/framework.htm

[4] Farnsworth, William, What Do I Put in a Security Policy, Aug 10, 2000; www.sans.org/infosecFAQ/policy/policy.htm

[5] Stallings, William, Network Security Essentials, p322-330.

[6] Integralis, System Security Services, Oct 20, 2000; www.integralis.ch/press/pr_014.php

[7] Timothy Turrell **,** IDS**,** September 13, 2000; www.sans.org/infosecFAQ/intrusion/IDS.htm

[8] Stallings, William, Network Security Essentials, p. 313.

[9] Denning, Dorothy, Information Warfare and Security, p. 321.