Global Information Assurance Certification Paper

ASP SECURITY; DEFENSE IN DEPTH

Dustin Osgood

GSEC – Securing an ASP; Defense in Depth. V1.4

TABLE OF CONTENTS

Abstract

This paper will address the task of securing Application Service Providers (ASP) using the Defense in Depth model. This topic will be addressed from three different perspectives – pessimistic, single point of security, and the practical view. A real world environment will further illustrate how there is only one view applicable for securing any information system. Arguments for each perspective will be presented and defeated. Arguments, evidence, reality, and common sense will ultimately persuade views in favor of the Defense in Depth model – the practical view.

I. Introduction

Defense in Depth, also known by some as layered security, is the most effective method for securing any type of information system. This is especially true in an ASP environment. At some point in their career, every security professional has been asked to secure an operating system, application, or database. After several rounds of research, modifications, and tests, the system is as secure as possible. A rookie security professional, or even some, seasoned professionals, may say to themselves, "Try to hack into my system." They couldn't be more wrong! They will quickly discover one of two things: 1. No one can use the system because almost all functionality has been turned off to secure it, or 2. The access points and/or functionality left open for those "necessary" business reasons are easily exploitable or have known vulnerabilities. Even the security measures put into place to compensate for the insecure functionality that has to be available is vulnerable. So, how does one secure systems without stripping them of all functionality or burying them six feet under? Defense in Depth will help us answer that question.

II. Body

Securing information systems can be analyzed from three different perspectives. The first being the pessimistic view, arguing that nothing can be secured 100% so why even attempt it. The supporting point is that there is documented proof that all security measures are exploitable, so what's the value proposition of attempting to secure any part of the environment? A second view focuses on a single point within the architecture for implementing the majority of the security controls. This is in hopes that a single point of security will be easier to maintain, thus able to avoid and block the majority of attacks. A third view is the positive/practical view, arguing that doing nothing is simply irresponsible and the primary objective of implementing security is to prevent both malicious and non-malicious incidents. The most proven security model, and best counter argument to all the other views, is Defense in Depth. The real world example will prove that the Defense in Depth model s the only way to achieve secure, operational

functionality while maintaining the functionality needed to achieve business needs.

PESSIMISTIC VIEW

Although this option may seem acceptable to some, it is considered by most as the least practical and irresponsible.

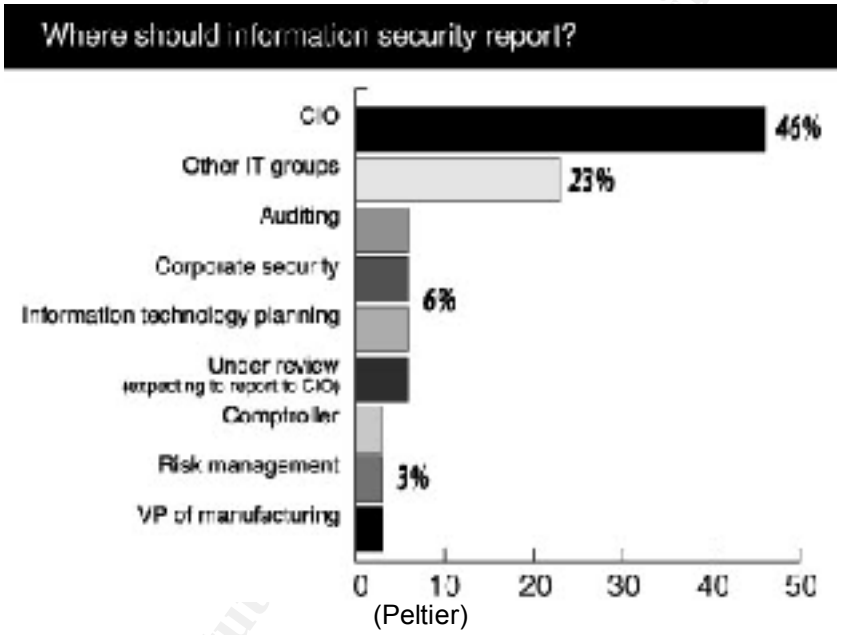
Unfortunately there are still IT and non-IT professionals that do not see the importance of security. It is obvious that nothing can be 100% secured and protected from all possible threats. However, just as the government continues to spend billions on defense mechanisms, so must security professionals continue to strive for complete security. In the month of July the CERT Coordination Center released twenty-two Vulnerability Notes and four Advisories (the CERT Coordination Center, http://www.cert.org/advisories/, 9/5/02). In the month of August they released twelve Vulnerability Notes and three Advisories (the CERT Coordination Center, http://www.cert.org/advisories/, 9/5/02). Of course these vulnerabilities and advisories span numerous operating systems, software, and platforms from various vendors, yet the importance of them cannot be discounted. Unfortunately, keeping up with this many vulnerabilities can quickly discourage a system administrator or security professional.

Simply addressing all the vulnerabilities does not secure an environment. It is well known that the largest percentage of attacks originate from within an organization. In fact, as reported in a study by Activis of 146 companies, 81% of security breaches originated internally, another 13% percent came from ex-employees and 6% from external hackers (Robb).

Regardless of security practices or controls, if the workforce is not properly trained and informed even the best security controls can be circumvented. Social engineering can be used against a company to either commit an attack or get one of the company's employees to commit the attack for the hacker. For example, a hacker could place a call acting as the lead internal IT person and ask the on-call employee to remove a directory on a file server because the information has been moved to a different directly and is no longer needed. The employee honors the request and an enormous amount of information is lost. As you can see, this act was not a malicious act by intent. However, because the employee was not properly trained follow authentication and change control procedures a great deal of business critical information may have been lost.

Given the evidence above it is obviously difficult to secure any environment, and maintaining a secure environment can be even more difficult. So why secure any part of the environment? Many management teams do not see the value in putting a lot of resources, human or monetary, towards securing and maintaining

security of their information systems for several reasons. These reasons vary from security not adding to the bottom line of the balance sheet to just plain ignorance of not understanding where security fits into the organization. The latter of these is more important than most executive realize. Personally, as a Security Architect I have reported to the Director of Operations and the Director of Engineering. Neither person has positioned the security practice to succeed. The security group must be in a position to independently audit the entire organization and enforce security policies. It is my personal belief that that security professionals report to the legal department or risk management, but most studies show that security organizations report to the CIO (Peltier).



(Peltier)

I believe that reporting to the CIO is a conflict of interest for two reasons. The first is that the user groups that should be monitored most attentively report to the CIO as well. Secondly, IT investigations and audits could produce results that have legal ramifications. Therefore, reporting the legal or risk management allows the security team direct access to the individuals capable of dealing with any issues. This could be very critical if there becomes and issue with the CIO.

FOCUSED SECURITY POINT

The limited amount of security resources, both monetary and human, drives many security professionals to choose an area they know best or one they have the most control over, and make it the focus of all security efforts. As a result of this narrow focus, other important technology layers or processes may be skipped or given too little attention.

There are five main areas where security can be addressed – physical, network, application, database, and operating system. There are different theories on which layer should be addressed first. Some would agree that focusing efforts on an outside layer such as physical or network makes the most sense because if hackers get past this layer, regardless of what is protected on the inside, they own the network and can shut down the business at will. The counter argument is that protecting the inside layers such as the operating system or database makes more sense because that is where business sensitive and critical data resides. The most critical layer is always evaluated differently, but a good indicator of the most critical point is to employ the 80/20 rule. That is, to determine where 80% of your attacks will come from and where the attacks target. For many companies this evaluation is getting simpler because so many companies have web servers. Incidents.org reports that HTTP, typically port 80, is the number one target of attacks. It is so heavily attacked that it is hit more than double the number of attack attempts than the next most attacked protocol (Internet Storm Center, http://isc.incidents.org/port_details.html?port=80, 9/5/02).

HTTP attacks for the last 10 days of August

Date	Sources	Targets	Records
2002-08-31	13156	193482	729552
2002-08-30	14861	188665	964416
2002-08-29	16483	145492	848394
2002-08-28	15767	176127	955043
2002-08-27	14538	125775	641285
2002-08-26	15067	116747	491596
2002-08-25	10920	95341	395937
2002-08-24	11392	61101	314168
2002-08-23	12383	44010	251638
2002-08-22	11559	86720	346495

(Internet Storm Center, http://isc.incidents.org/port_details.html?port=80, 9/4/02)

In addition, Microsoft released a cumulative patch related to their web server software, Microsoft Internet information Server (IIS), to cover eight different vulnerabilities.

So what's the problem with blocking 80% of all attacks? The obvious argument is that it only takes one incident, but an equally true argument and possibly a more business appropriate argument, is that this method leaves a single point of failure. When evaluating and incorporating business needs you will open holes in the focus point, ultimately exposing the rest of the environment where security

controls are basically non-existent. Precisely the reason for employing the "Defense in Depth" model, also known as "Layered Security."

PRACTICAL VIEW

Pessimists are right, nothing can be 100% secured, so security professionals have to find the best method to secure the *entire* environment. Defense in Depth, is the most comprehensive approach to addressing security needs and understanding risk management.

As mentioned previously, there are five areas where security can be addressed – physical, network, application, database, and operating system. When secured properly, each layer helps strengthen the security of the overall environment.

Let's begin with the physical layer. With proper controls in place such as controlled access to facilities, locked entry points, and environmental protective measurements, the physical layer can protect computer systems from some unauthorized access and environmental threats. Although the primary purpose of physical security is to protect assets from environmental threats, it also compliments other layers by safeguarding personnel and limiting unauthorized access threats (Roper 1). The obvious vulnerability here is often related to having visitors within the facility. Visitors should be escorted at all times to not only protect physical assets, but also to protect intellectual assets and financial information.

The network layer is the outer shell from a technology perspective. This layer and the database layer are often considered the most complimentary to the other layers and to the entire environment. The network's primary purpose, of course, is to facilitate communication between computer systems. However, this inherently acts as a security control by allowing only certain protocols, defining traffic as one or two-way, and implementing ACLs (Access Control List). This obviously protects systems from traffic coming from unauthorized sources and traffic not being transmitted using the allowed protocols. However, if the network is compromised it is easy to discover all the information systems in the environment. In addition, the controls put in place can obviously be modified or simply removed. Many hackers will redirect traffic to another destination or define an additional route to allow themselves access.

The application layer may or may not be considered the next sequential layer, but the application interface is often the most accessible once in the network. In most applications the data it presents is separate from the application itself. This, of course, is a good thing because segregation allows better access control and more control points. Most application security is implemented into user access privileges and roles. This controls access to the actual data that can be

viewed and possibly changed, reducing the need to login directly to the database. A secure application is purely an interface to the data and does not allow a user to have overlapping role responsibilities. This leads to a very common problem with applications – users can often have many roles assigned to their ID, allowing them the ability to perform tasks that should be segregated. A separate problem is that of an employee's job position requiring them to have conflicting job responsibilities. A prime example of this is in the SAP application. During a segregation of duties testing that I assisted in, several instances were found where users were allowed to both create and approve purchase orders. The employees' positions did not require this, but the roles assigned to them within the application allowed conflicting privileges. Within these roles there were legitimate privileges that employees needed to perform their jobs. However, several privileges are lumped into each role, it resulted in a segregation of duties issue. The other very common application vulnerabilities are buffer overflows and privilege escalations, often the former resulting in the latter. So, although the actual data is somewhat protected by the application, it is obvious that not only is there a need to directly access the data, but also that the application cannot provide the necessary security controls even if direct access to the data was not needed.

The database is where the critical business data lives, so the need for security controls is obvious. This layer, as mentioned previously, is often considered the next beneficial to the rest of the environment. Only database administrators should be allowed to login directly to the database. Proper database security is extremely crucial because even if the network and application are compromised, the company's sensitive data cannot be easily altered. However, the reason I feel the network layer is just as important as the database layer is because if the network is compromised user ID's can be easily discovered; therefore, obtaining access to the database is only a matter of time. Database vulnerabilities are equally as common, but the most common are buffer overflows caused by large queries. For example, the CERT Coordination Center released five Advisories related to SQL Server where four of the five were buffer overflow vulnerabilities (The CERT Coordination Center, http://www.cert.org/advisories/CA-2002-22.html, 9/5/02).

Now that it has been demonstrated that all the previous layers have the potential to be compromised, we are down to the last layer of defense, the operating system. The operating system is often referred to as the "keys to the kingdom." Regardless of whether it is the application, database, or network device, they all run on an operating system. If the operating system is compromised then everything running on that server can be compromised as well. However, with proper controls in place the operating system can protect information systems from being compromised. Access privileges, logging, configuration management, and only enabling the necessary services are just a few of the

more important controls. Default installs of operating systems are the most vulnerable operating systems (SANS Institute). Many vulnerabilities exploit privilege levels or allow intruders to gain administrator level access.

Again, the pessimists have proven their point – nothing can be 100% secured. Every layer can be compromised either by a known vulnerability or by social engineering. However, Defense in Depth can help address the different vulnerabilities at each layer. As shown, each layer can help fill the gaps opened by other layers and allow business requirements to be implemented.

REAL WORLD – ASP SECURITY

An application service provider (ASP) could not survive without the Defense in Depth model. Security concerns around hosted applications result from two inherent components of the ASP model: 1. The application is installed in a remote facility that is not owned, managed, or protected by the customer's employees. 2. All access to hosted applications occurs over the Internet (Kelman). Concerns rising from these inherent model characteristics include loss of critical data, theft of data by competitors, and loss of privacy of confidential information (Kelman). Possibly the most important rule, Kelman states, is that the ASP must isolate each customer's data. Based on personal experience, the scope of a typical ASP may include support for six different enterprise applications, two database vendors, three operating system vendors, two web server vendors, numerous network devices, and backup and storage services. Also, every customer has unique architecture wishes. Customer A may run PeopleSoft on SQL Server database on Windows 2000, and Customer B may run PeopleSoft on Oracle database on Solaris 8. Furthermore, access privileges have to be defined for each of the support groups as well as customer users. Along with each of the approximately fifty customers having different architecture requirements, they each have different business needs. Obviously addressing security on a customer-by-customer basis is not feasible. The first step is to develop security standards and policies, then plan for exceptions. On each new environment built the standard must be followed. Not only do the standards allow for quick building of the environment, but it also gives the security team a baseline to understand what the environment looks like.

Once a customer begins introducing customizations the environment may become less secure. However, this is where Defense in Depth will allow you the flexibility to meet and implement customer requests with little sacrifices to security.

BUSINESS CASE 1

Company A would like to open FTP (file transfer protocol) from their internal network to their environment hosted in the ASP's data center. There are a few issues with enabling FTP. First, it is the third most attacked port as reported by the Internet Storm Center (http://isc.incidents.org/, 9/5/02). Secondly, there are many vulnerabilities associated with FTP if it is not implemented correctly, namely the ability to login anonymously. To be more exact, there are eleven vulnerabilities reported spanning various FTP products (Internet Storm Center, http://isc.incidents.org/port_details.html?port=21, 9/5/02). Third, and possibly the most important, FTP transmissions are in clear text. So what security controls are in place to allow this customer request to be implemented? Customer A has a private connection into the data center instead of a VPN; therefore, the traffic will not be in clear text across the Internet. Although many people expect that a VPN is always encrypted, its not. Also, the firewall can be configured to only allow FTP connections from a specific IP address to a specific IP address and is only allowed inbound. Defining allowed FTP sources greatly reduces the threat of FTP being attacked at a high rate. There may continue to be numerous attack attempts, but because the firewall has been configured correctly all traffic not originating from the specified IP address will be dropped. As well, if the host is compromised and the intruder attempts to initiate an FTP session outbound it will not be allowed because of the rule specifies that only inbound FTP sessions are allowed. Finally, there will only be one user that has knowledge of the FTP account login information, greatly enhancing accountability capabilities. As you can see in this example the network layer was used to ensure that traffic originated from a trusted source, as well as controlled allowed traffic direction. Only the necessary services, in this case FTP, were enabled at both the operating system and network layers.

BUSINESS CASE 2

Customer B would like to introduce some new functionality into the application, which requires a generic ID to be created at the database layer to allow the application to login to the database. The biggest problem with generic accounts is the loss of accountability. In this situation the account logs into the database, which we already discussed increases the chances of data being modified by unauthorized users. Futhermore, this ID must have administrative level privileges within the database in order for the new application functionality to work. In this case controls can be put in place at both the application and database layers. At the application layer do not allow users to login with the generic account, or switch to the generic account, to perform work. This will increase accountability capabilities. At the database layer users should not be allowed to login directly with the account, and if the account uses a password then a limited number of people should have knowledge of the password. This also increases accountability capabilities. Furthermore, strong password controls should be implemented including the requirement of special characters,

numbers, letters, and at least ten characters in length. In the event that a hacker attempts to crack the password it will be much more difficult to accomplish. This becomes extremely important because this account, as mentioned, has administrative privileges within the database.

As you can see from the two business cases security controls are being modified or factors that are known to be risks are allowed in the environment. More often than not, security work is about managing risks and mitigating those risks. There will always be risks associated with every piece of functionality enabled. The key is to manage those risks so that when the environment is exploited there is little exposure to the rest of the environment due to the security controls in place at all layers. Defense in Depth allows risks to be more easily managed because if risk is introduced into one layer it can be mitigated at another layer.

III. Conclusion

There are always options, but even with limited resources the best option for securing an enterprise's information systems, especially in an ASP environment, is Defense in Depth. In all of the five main areas where security controls can be implemented it was demonstrated that they each have the potential to be compromised. Each layer can be considered equally important to the overall strength of the environment. What one layer cannot control can often be accounted for by another layer. This is the key to the Defense in Depth model. There will always be a need for customizations to information systems, no two companies will use the same software exactly the same. So when an ASP attempts to not only secure their own information systems, they must also secure many other companies' information systems. With the customizations comes a daunting task of finding a way to secure each information system. This would no doubt be nearly impossible without following the Defense in Depth model.

WORKS CITED

1. Activis. 5 Sept. 2002. <http://www.activis.com/en/>.

2. The CERT Coordination Center. 5 Sept. 2002 <http://www.cert.org/advisories/CA-2002-22.html>.

3. The CERT Coordination Center. 5 Sept. 2002 <http://www.cert.org/advisories/>.

4. E-Business Advisor. 5 Sept. 2002 <http://advisor.com/adv/AdvisorHome>.

5. Internet Storm Center. 5 Sept. 2002 <http://isc.incidents.org/port_details.html?port=80>.

6. Internet Storm Center. 5 Sept. 2002 <http://isc.incidents.org/port_details.html?port=21>.

7. Internet Storm Center. 5 Sept. 2002 <http://isc.incidents.org/>.

8. Kelman, Ariel. ASP Planning: A Checklist for Security. 10 May 2001 <http://advisor.com/Articles.nsf/aidp/KELMA02>.

9. Peltier, Thomas R. (1997) 6 Sept. 2002. <http://www.gocsi.com/infopro.htm>.

10. Robb, Drew. (15 July 2002) 5 Sept. 2002. <http://www.esecurityplanet.com/trends/article/0,,10751_1405031,00.html>.

11. Roper, C. A. *Physical Security and the Inspection Process*. Butterworth-Heinemann, 1997.

12. SANS Institute. *The Twenty Most Critical Internet Security Vulnerabilities; The Experts' Consensus* Version 2.502. 30 Jan. 2002.