Global Information Assurance Certification Paper

Paul R Buzzell

GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4b (amended August 29, 2002)

Enhancing risk management within a research laboratory, from behind an academic institution's firewall – a case study

Abstract:

My laboratory is located at a state university. I implement experimental designs in several areas of Internet-based behavior research. This includes interpreting grants, creating applications corresponding to components outlined in the grants, and assembling these applications into a site that will then record all potential outcome measures with emphasis on the primary outcome measures. In the actual creation of sites, the only help I've had is from graphic designers, and eventual feedback from beta testers. These studies require interaction with people outside the campus area network. Security had never been in the job description of anyone associated with the laboratory. Because of this, the only security measures taken to protect the two Windows 2000 servers were virus protection and frequent patch application to the operating systems as well as fixes for the ColdFusion server software I use. (http://www.macromedia.com/support/coldfusion/ts/documents/tn17254.htm) Additionally, there was no university-wide protection in the form of a firewall.

In 2001 and 2002, network and computer security awareness was heightened at the university through several incidents affecting the institution's reputation and/or monopolizing the time of information technology personnel. Most of these incidents were a result of mass-distributed malware – primarily worms. By the end of 2002, a firewall was in place and any servers on campus that were not explicitly owned by the university had to meet several criterion and be approved by the university's security team in order to maintain their connection with off-campus users. It was at this time that my department and grant writers were forced to commit resources for securing our servers. After my responsibilities were changed, giving me time to manage the security of the systems in my lab, many risks were brought to light, then mitigated, transferred, or accepted. Much of this work was made possible by the department's willingness to invest in this course for me.

Introduction:

There are many fields of applied research involving the Internet and human subjects; I have been implemented experimental designs in a few. The commonalities among these fields include many concerns -- not the least of which is providing a confidential and readily accessible environment for the exchange of information among participants and researchers. Also, since the locations of such research systems are typically centered in academic institutions, there are additional risks that must be managed. There is a belief within academia that an open policy is essential to the advancement of our

society. The need for such advancements and their link to educational institutions are outlined in this quote from a judge: “Teachers and students must always remain free to inquire, to study and to evaluate, to gain new maturity and understanding; otherwise our civilization will stagnate and die.” (354 U.S. 234 (1957)), and this quote from the Electronic Privacy Information Center (EPIC): “Monitoring the content of communications is fundamentally incompatible with the mission of educational institutions to foster critical thinking and exploration.” (http://www.epic.org/privacy/student/p2pletter.html)

Because of the need for an open policy and the greater challenges such a policy provides to network security, academic research institutions have been viewed as easy targets for attacks. Whether or not the state of an institution’s security allows for easy intrusion is, in fact, not as important as the perception – which, in itself, can increase risk. More recently, however, many academic institutions have attained a heightened awareness of security risks in today’s networking environment and are tightening network policies and implementing firewalls with more and more restrictions.

The challenge now presented to Internet-based behavior researchers is to effectively explore interactions between people via the Internet without allowing tighter restrictions or growing risks to impede experimental designs.

Before:

At the institution where I am employed, a firewall system was put in place between 2001 and 2002. University policy requires certain criteria to be met prior to issuing a firewall waiver for a server. A waiver effectively allows an opening in the firewall for outside access to and from a server behind the firewall. The main reason for restricting what machines are accessible from outside the campus area network, of course, is to mitigate risks associated with vulnerabilities in machines behind the firewall. These vulnerabilities could be used as avenues of attack by malicious users or their software thereby jeopardizing the security of the entire campus area network.

In our case, the three criterion we had to meet to the satisfaction of the university security team in order to be granted a waiver were:

1) Structured education of an assigned system administrator.
2) Establish a development server system
3) Compliance with one of the university classes of servers (‘Web only’, ‘Web and email’, etc., with well documented reasoning for any straying from these classifications).

Concomitant to working toward meeting the university’s requirements, I looked into other potential threat vectors. Vulnerabilities from on campus as well as Web page security were risks I deemed to be foremost among potential threats to the information and systems I oversee. However, with no waiver, a lack of access to our Internet-based intervention programs would override any other concerns.

If no tools need access from outside an autonomous area, security would be much easier. Unfortunately, among the tools required to be available to

participants in these studies are interactive dynamic Web content, chat, and email. The objectives of many studies we are currently conducting include simulating face-to-face interaction. Ideally, cutting-edge chat-type applications would be available to help simulate face-to-face meetings. Also required is the ability to track individual participants' usage of yet to be determined site components as well as usage by treatment group. In addition, given the reliance upon volunteer participants – many with below average computer skills and varying types of connections, there was a need for very simple means of connecting to the servers with a minimal requirement for the downloading of software. The population sample, combined with the lack of University affiliation ruled out virtual private networking as a means of working around the firewall.

The resources at our disposal are sparse. Available are two production servers running Windows 2000 and IIS5 – one of which is nothing more than an old Pentium III desktop; one other desktop machine running Windows 2000; Macromedia software including Coldfusion server and Flash (http://macromedia.com/); Authentix software by Flicks (http://www.flicks.com/flicks/authx.htm); and little to no immediate funding.

During:

A business impact analysis was not required to determine that obtaining a waiver was priority number one. The first hurdle in obtaining said waiver – item one in the above list – was to assign someone the responsibility of managing risks to these systems and enroll them, as soon as possible, in a security curriculum that met the university security team's approval. Since I am the only full-time employee working with the servers, I was assigned this ongoing task. And it was determined by the university's security team that completion of this course would be sufficient structured education, when combined with assurance that the material would be put to practical use, enrolling me in the SANS GIAC Security Essentials course met one condition for the issuing of a firewall waiver for the laboratory.

Also among the conditions for the issuing of a waiver was the use of a development server for the testing of new software, new releases of old software, different settings, new patches, policies, etc. For this reason, one of the first steps taken in securing this research laboratory was also crucial in initiating the waiver issuance process. The spare desktop computer was designated as a development server to help mitigate the risk of inadvertent administrative denial of service 'attacks.' For example, while testing new policies, installing patches, etc., there is frequently a need to restart the machine more frequently than would be advisable for a production server. Also, with a development/testing machine, the administrator has the ability to be more progressive with testing potentially detrimental settings prior to considering implementing such settings on a production server; thereby facilitating and augmenting the familiarization process with security-related issues that can be applied to the production servers without negative effects.

In contrast with what the institution wanted in item three above – all servers to comply with one of their standard classes – our research study

application requirements did not fit one of their categories. Our requirements did not match due to the services in use at the time. The tools needed for our research projects had to be modified in such a way that they would no longer require certain services not included in one of the institution's categories of server. One might argue that these tight restrictions on common services such as real-time chats could affect the validity of our studies; many of which rely on ready communication among online participants. Without state-of-the-art communication tools, are we truly testing the capability of the Internet as an intervention tool? Nonetheless, at the time, we were in no position to argue the point and since such a point is beyond the scope of this article, I will not pursue it further. We decided to comply with the university's wishes and sacrifice performance.

Having complied with the first two items on the security team's list, (1. obtain structured education for an assigned system administrator; and 2. establish a development server system), I then began exploring ways in which to change the production systems themselves to subsequently increase our chances of obtaining waivers. Although not the only determining factor, the more ports requiring access through the firewall, the smaller the chance of a waiver being issued – to say nothing of the greater risk to vital information. The two servers, requiring multiple ports to be open were viewed as a hazard to the university. For this reason, along with the knowledge that there was nobody responsible for managing security risks on these machines, the prospect of continuing to allow these machines interaction with outside users presented enough risk to result in a prognosis of 'doubtful' that a waiver would be granted, according to the university's security team.

However, great progress was made in helping the systems comply with a university classification of server that was more likely to be granted a waiver. Initially, one server needed a port open for chat as well as port 80 (for Web traffic). The other server needed four ports open for functionality of its existing email applications (25 for SMTP, 106 for Password Server, 110 for POP3, 143 for IMAP4) as well as an additional port (80) for Web traffic. The chat application was also an expense we could not sustain indefinitely. For this reason, I decided to create an html chat application using ColdFusion and Flash. This chat application has now fully replaced the original chat application, it is free, more flexible in meeting our needs, and uses port 80 exclusively. For now, the delay associated with html chat is disguised by adding a user's comments to their chat interface before the database is updated and re-polled by the user's machine, thus tricking the users into thinking it is a 'real-time' chat. This functional html chat eliminated the need for one of the seven ports initially needed.

Email access was built into one of the currently active online studies. All participants in that study were automatically given an email account upon completion of a baseline questionnaire. It was determined that the needs of the study could be met with a Web interface for accessing and sending mail, and that there was no need for attachments. At this point, I decided to look into transferring some risk to the university email system as well. With the help of network administrators, emails sent to our email server from outside our

autonomous system are now intercepted by the university's gateway server, forwarded to their email system, scanned and stripped of attachments, then forwarded to our email server from behind the firewall. Concurrently, I was creating a Web interface to be placed on the server hosting chat and the majority of Web content (not the email server). This interface collects information from the email server and displays the results for the participant. Since the email interface was created in Flash MX and users in some studies had the freedom to upgrade their Flash Player to version 6,0,47,0, the email server had to be added to the 'sandbox' within the Flash movie by using the System.security.allowDomain function.
(http://www.macromedia.com/support/flash/action_scripts/actionscript_dictionary/actionscript_dictionary721.html)

Prior to the release of Flash Player 6,0,47,0, there was no need to include this function, however, this is because prior to Flash Player 6,0,47,0, there were no restrictions on the Flash Player's ability to exchange information with any domain – even the local machine. At the time of the release of this new player, we were conducting a study that strove, in part, to simulate broadband access for participants throughout the state. I had developed an all-Flash site that relied on ColdFusion scripting at the server end to load and exchange variables with Flash files on the local machines, and we were days away from launching this site and weeks away from launching a similar site for home users when the new player was released. Obviously, the new Flash Player is a necessary security feature, but in the case of the machines at our research test site, the resulting inability to access content demonstrates that an enhancement for some can be detrimental to others; naturally, with the new Player, no Flash content could be loaded and the site was totally non-functional until we manually installed the Flash Player files from a previous Player. If I had participated in the Security Essentials course prior to helping with the experimental design of that particular study, the potential for a malicious Web developer exploiting the ability to access local files would have been more likely to occur to me. I, therefore, might have foreseen a patch for a feature I had never considered a vulnerability, and intervened at the grant-writing stage before an entire site was developed and could not be used without leaving a potential vulnerability on the machines being used by participants.

One workaround we found for these sites relied on the ability to point to local files from the Web – essentially using cross-protocol linking, pointing to a 'file://….' address. After misleading myself once by not considering a feature a potential vulnerability, I decided against using any local files in the study relying upon the use of participants' home computers. And, shortly following Macromedia's release of their new player, Microsoft released Internet Explorer 6, Service Pack 1, which no longer permits Web content to point to or include content from the local machine. Of course, this would have foiled another aspect of an experimental design.

As an aside, it should be noted that those using an Internet Explorer prior to Internet Explorer 6, Service Pack 1 or Flash Player prior to Flash Player 6,0,47,0 should upgrade to help prevent easy access to content on their local machine while browsing the Internet.

Macromedia's System.security.allowDomain function is currently used to include other domains in the 'sandbox', however, it will not include the 'local domain' unless the Flash movie itself is local. Nonetheless, the System.security.allowDomain function was the key in allowing an easy, application-level fix to be implemented and thereby mitigate the risk of having more ports open than would be otherwise necessary for the email issue in my laboratory.

These steps, along with the fact that participants sending messages do not need any ports open to them directly since the server is doing the sending via the university SMTP gateway, allowed me to eliminate the need for any email ports to be open through the firewall. We were then down to two open ports – both servers' port 80 (for Web traffic). After moving any essential Web content to what is now the 'main' server, there was only a need for one open port among all the services and between the two servers!

It must be said that prior to participating in this course, the idea of limiting ports in this manner would never have occurred to me. While we still might have been granted a waiver, our systems would have been more vulnerable to attacks. The university was pleasantly surprised, and granted a waiver promptly.

The threats presented to the university network by the servers were unknown as were the threats to the information being collected and stored on the two systems. Initially, these issues were set aside since foremost among the concerns was meeting university requirements and obtaining waivers for the servers – without which we could not continue to operate. For this reason, priorities were slightly different than if security had been deemed more important than the continuation of ongoing studies and maintaining of funding.

Obviously, obtaining a waiver was foremost of my concerns. Alternately, security issues that were not seen as immediate obstacles, while not focused on with nearly as much urgency, were looked into even before a waiver was granted. These included port scans, dynamic Web page script validation, backup scheme, policy and security settings.

Bare in mind that the aforementioned security issues focus on risks that can be protected against at the application layer of the TCP/IP Protocol Stack since my research laboratory does not have network administration rights or responsibilities. Because of this, there are limited options for implementing direct security measures below the application layer.

Once the waiver was issued, I could focus more on other security issues that would protect us from attacks from within the campus area network. My first steps included port scans followed by disabling or uninstalling any unneeded services. After experimenting with LANGuard (http://www.gfi.com/languard/) and NmapWin (http://www.insecure.org/nmap/nmap_download.html) an accurate picture of some existing vulnerabilities developed.

Also, to help with information assurance, a backup system was put in place and tested. After review of several rotation options for backup, a somewhat non-traditional system was established to minimize cost and risk of data loss. Following consolidation of critical files in current use, the tower of Hanoi (http://www.certance.com/products/cartridgeRotation/cartridgeRotation.html)

method was selected to be integral in the backup plan – alternately using tapes and a hard drive in a full backup rotation scheme. The most recent tape is stored off site for further protection of the information. Also, the hard drive is used as 'media set A' in the rotation scheme making it the media for use every other day. This variety of media could be beneficial if one media type fails, in which case there will be another type of media with information only one day older.

In addition, there were a number of potential vulnerabilities in a number of Web applications that needed to be addressed. Among these vulnerabilities were several ColdFusion errors that could be elicited to give a malicious user clues as to how to bypass measures intended to control information access authorization. For the most part, these error messages were displayed when a variable was not defined – either not entered in a form, or lost when a session timed out or a cookie was deleted. After several days of watching the ColdFusion server's application logs and editing all pages that elicited an error message, all errors that indicated potential threats stopped. This was accomplished by redirecting users whenever a needed variable was not defined or setting a variable to an acceptable value, if not previously defined.

All sites available to study participants in our laboratory are password protected. The reason for this is twofold: 1) confidentiality of participant information 2) tracking use of site components by user is an essential data collection method.

Establishing a seamless logon system using Authentix and ColdFusion session variables was a challenge. Authentix can be configured to set a session cookie upon logon and is essential for tracking Web usage to the extent required for these studies. Unfortunately, the variables set by Authentix are not compatible with the ColdFusion scripting used throughout the remainder of all the password-protected sites in the laboratory. Having users enter their logon information twice was not looked upon as user-friendly or aesthetically pleasing. For this reason, functionality was given priority over security. Once the user was authenticated via password authentication and access to the site-in-question's folder was authorized, the username was passed in the URL to a ColdFusion page that in turn set the ColdFusion session variables. The username was the only authentication prior to authorizing assess to specific data within the site. The action occurred in a very obscure manner, but to paraphrase from this course's material: 'security by obscurity is not security.'

It was possible for someone to log on as one user and access content meant for another user. By submitting a query string to a hidden file within the protected folder while still logged onto the site and submitting another user's username (and other data) with the string, one user could assess another's data. The risk of this happening was considered miniscule, due to low apparent threat, given the lack of motive on the part of study participants, the lack of technical know-how, and the obscurity of the current process (the page to edit was never displayed or cached and has an unlikely name and location). Nonetheless, the vulnerability was too much for me to tolerate long-term.

The solution had three components: 1) After authentication via Authentix, but before authorization, the logon variables are set as ColdFusion session

variables. 2) The session variables were then sent with a token when the user is forwarded to the site-in-question. 3) The script previously looking for a username in the URL now looks for all logon information in a session cookie. The data are validated by the server and users are then forwarded elsewhere when data are found to be missing or incompatible.

I used the Center for Internet Security's scoring tool (a best practices risk assessment tool) to evaluate local policy settings on all three servers – the development server, the email server, and the primary production server. Since the vast majority of settings assessed by the CIS-Win tool had been left as their default values, all three machines had very low scores initially (less than 3.0!). While going into the details of how potential risks found with the CIS scoring tool were mitigated is beyond the intended scope of this paper, let it suffice to say that the tool was extremely valuable, if a little vague at times, in pointing out vulnerabilities and solutions. However, the SANS content did help point me in the right direction in correcting vulnerabilities (SANS. "Risk Management and Auditing." Unit 3,6 of SANS Security Essentials).

After:

Research in my laboratory is ongoing. This is not an outcome that was seen as likely in the eyes of the university where I work, but it certainly attests to the effective enhancements to security I was able to implement since the onset of the university's firewall project. Not having been issued a waiver for the server would have resulted in a denial of service state that could very well have closed the laboratory. Continuation of research in this laboratory would not have been possible without information and concepts taught in the SANS Security Essentials course. While much of the content consisted of crucial details specific to a given operating system or situation, the most valuable material learned through participation in the course was a heightened awareness of potential threats and threat vectors. Such learning is essential in effectively managing current risks as well as improving one's chance of effectively mitigating future risks. Many of the actions taken were a result of impressions made on me throughout the course.

In the process of meeting the requirements for the firewall waiver, security was enhanced as well. A development server not only adds another line of defense during times of change, it has also become another server we could use as a production server in emergency situations such as total failure of the primary production server – part of a continuity plan. Without training in the field of security, I would not have been able to mitigate risks nearly as effectively as I have. I also would not have considered transferring risk as I did in resolving the email receiving issue. More importantly, I would have been oblivious to risks that are now apparent to me. Make no mistake, there are still vulnerabilities on these systems. It is only a matter of time before the next notice of an operating system-related buffer overflow is uncovered. There are also opportunities for participants to access information within their site outside the prescribed schedule indicated in their studies' experimental designs. Undoubtedly, there are also other vulnerabilities I have never considered. Given the basic concepts and resources

introduced in this course, I am more confident in my ability to manage risks to these systems and to deal with any incidents that may occur.

Among the most prominent risks currently is a lack of policy in written form. My 'to do' list contains 'write job description/security policy.' Prior to writing a new job description, some other policies need to be addressed, however. One of which is the amount of time we are willing to commit to security. If this proves to be all of my time, the job description will be simpler, and the security policy can be more strict. Since these policies that must be set first are out of my control, I've focused on other issues in the meantime.

One such issue revolves around logging. I currently use netstat, Event Viewer, ColdFusion server logs, and IIS logs, but not systematically. The lack of an automated intrusion detection system is a potential detriment to security, but until more funding is available, my manual checking of activities will have to suffice.

Finally, plans for future research would require the installation of software capable of 'pushing' data to participant-users. The software package we are considering is Macromedia's Communication Server. The first hurdle to overcome would be to be granted approval from the university to have more ports open, or more ideally, another server with three ports open through the firewall. There will be other issues we will have to address during the testing stage, before introducing this new tool to a production server (http://www.macromedia.com/devnet/mx/flashcom/articles/security_overview.html). The reasons for our interest in this software, aside from improving our chat capabilities, include evaluating online gaming applications as education instruments.

All in all, this curriculum has been very helpful to me. The laboratory is more secure, the research is proceeding seamlessly, and I am more confident in the integrity of our data.

References:

Certance LLC. “Cartridge Rotation Schemes.” URL:
http://www.certance.com/products/cartridgeRotation/cartridgeRotation.html
(22 Apr. 2003).

Macromedia, Inc. “System.security.allowDomain” Macromedia Flash Support Center, ActionScript Dictionary. URL:
http://www.macromedia.com/support/flash/action_scripts/actionscript_dictionary/actionscript_dictionary721.html
(22 Apr. 2003).

Macromedia, Inc. “Security Best Practice: Securing the ColdFusion Administrator.” Macromedia ColdFusion Support Center, TechNote. 19 May 1999. URL:
http://www.macromedia.com/support/coldfusion/ts/documents/tn17254.htm
(22 Apr. 2003).

Rotenberg,Marc, Hoofnagle,Chris Kessel,Adam, Agrawal,Ruchika. Electronic Privacy Information Center. URL:
http://www.epic.org/privacy/student/p2pletter.html
(22 Apr. 2003).

SANS. “Risk Management and Auditing.” Unit 3,6 of SANS Security Essentials 1.8v

Simmons, David. “Macromedia Flash Communication Server MX security overview.” URL:
http://www.macromedia.com/devnet/mx/flashcom/articles/security_overview.html
(22 Apr. 2003).

Warren, Earl, Chief Justice. “Sweezy v. New Hampshire.” 354 U.S. 234 (1957)