Interested in learning more about security?

SANS Institute InfoSec Reading Room

Open Source Risk Mitigation Process

The Open Source Risk Mitigation Process described in the previous pages is a tool for corporations to use when trying to understand why a simple decision to use the "free" Open Source software should be taken very seriously. The ramifications of using just any Open Source Solution without understanding its weaknesses and not implementing compensating controls could be disastrous and much more expensive than a purchased alternative. In some cases, Open Source Solutions fulfill all the requirements with few issues that n...

SANS Security Essentials
GSEC Practical Assignment
Version 1.4b
Option 2

Open Source Risk Mitigation Process

by
Carlos Casanova

I Introduction

In today's marketplace, everyone is working to get more from less while not compromising quality or security. This is not an easy thing to accomplish but given the revolution which Open Source software has brought, some of this may actually be realized in the coming years. The popular growth and acceptance over the past few years of Linux and the Apache HTTP Server combined with continued pressure to reduce IT expenditures and capital investments has some senior executives finally beginning to accept Open Source solutions as viable options to achieve their overall corporate goals. The adoption of Open Source solutions within a large international corporation isn't as simple as just downloading the software from a web site. There are many elements that must be researched and weighed to determine whether or not this is an avenue which a corporation should go in.

Many times when people speak about Open Source software, they only think of pieces of work like Linux and Apache HTTP Server but there are many other Open Source solutions which are not as large or mature. Some solutions are released by the same mature communities that released Linux and Apache HTTP Server but others do not yet project the same sense of comfort and longevity that The Apache Software Foundation (http://www.apache.org) does today. Regardless however of who releases the Open Source solution, the question remains the same, and that is, whether or not to use Open Source Solutions in production systems. The questions and research must address the issues of Security, Total Cost of Ownership (TCO), Licensing and Break/Fix before any recommendation can be made to use Open Source software. The hope is that these questions are asked and research is done before the answers or results are found to be unacceptable.

II The Problem

In today's Information Technology departments lower level managers are being pressured by their business clients and senior technical management to deliver quality production systems faster and at lower costs in order to meet Service Level Agreements (SLA) that they have agreed to with their business clients. In addition, Return On Investment (ROI) and TCO metrics must be also be met to achieve Information Technology Departmental objectives. To meet these demands some departments and their technical staff began looking to what was considered unconventional sources as a means to deliver the quality production systems and achieve all their other organizational requirements. The "unconventional" source that was pursued was to begin using Open Source software as a component of their new development efforts.

The decision to begin using Open Source software may appear to be an obvious choice based on the previously described demands but there are ramifications of such a decision. First, does the use of Open Source software violate the Corporate IT Policies & Standards that your company established for internally developed or purchased systems? By using an Open Source solution, have you invited copyright infringement or fair use litigation because you have not properly acknowledged the use of Open Source software in the development of your system? Are their royalties that would need to be

paid to the community if you use your system as part of a revenue generating process? Have you exposed your corporation to security related weaknesses which reside in the Open Source solution? How you will handle break/fix issues once your system is in production since there is no vendor under contract to provide support? Does the cost of implementing checks and balances around the above topics eliminate the savings of using an Open Source or "free" solution? It is very clear that making a decision to use Open Source is simple but the impact can be devastating if it is not done properly and with the necessary due diligence.

Open Source software carries with it many benefits potentially but it also carries licensing requirements and the potential to be used as a tool in compromising a corporation's network. There is tremendous debate whether Open Source software is less prone to security exposures because many "neutral" eyes review the code on a regular basis or if it is more susceptible because anyone with the time and energy can become intimately familiar with the software that created or makes up the inner workings of your corporation's deployed system. This paper will not try to answer the question of whether it is more or less secure because that debate is a philosophical one and has not been scientifically studied to establish metrics. However, the debate does bring to light that a company must implement some risk mitigation measures to address the fact that strangers, without any allegiance to or contract with your company will have a major role in delivering a secure system out to production.

Balancing the needs of the company to build systems in order to be able to deliver products & services while maintaining a secure, affordable & dependable environment is not as simple as changing the wording on a corporate policy or standard. It requires an in depth focused research effort into all aspects of the problem in order to fully understand how to solve it. When the policy violation was brought to the Information Technology Policy & Ethics Board (ITPEB) of my company, they determined that instead of simply reprimanding the department for the policy violation, they would establish a committee to review the existing policy(ies) and make a recommendation on whether or not the policy(ies) should be changed to accommodate the use of Open Source software in the future. If the recommendation by this committee were to be that the policy(ies) should be modified, the recommendation would be accompanied by a formal risk mitigation process which would address at a minimum the Legal and Security concerns expressed by senior management and the policy & ethics board members.

III The Approach

As a result of the decision by the ITPEB, a champion was chosen by the ITPEB from the Enterprise Architecture organization to assemble the Open Source Steering Committee (OSSC) to research and recommend how the corporation should proceed with the use of Open Source software. The committee was comprised of 4 sub-committees plus the Director and Lead Architect from the development team that brought the issue to the ITPEB. The champion decided to choose five people to act as the chairperson's for the different sub-committees. Four of the five sub-committees would address the aspects of the problem described above (Legal, Security, TCO and Break/Fix). The fifth sub-committee would focus on defining a process and system that

could track usage, deploy software and provide a notification mechanism for all users of the Approved Open Source solutions within the organization.

The leaders of each sub-committee decided that each of them would develop a recommendation and control process for their respective areas and present that to the OSSC for group discussion. Once all five recommendations and processes were discussed and approved by the committee they would be incorporated into one overall process document called the “Open Source Risk Mitigation Process”. The risk mitigation process document would accompany the formal recommendation that was to be delivered to the ITPEB. As with any new process or effort it was important to clearly define and document its goal and purpose, in this case it was stated as “To enable the corporation the ability to use the most cost effective software tools available while identifying, documenting and mitigating the risks associated with the use of those tools". The next section will describe some of the steps taken and thought processes associated with each of the five sub-committees. Four of the five sub-committees will be covered at a higher level to provide background and approach. The “Solution” section for the fifth sub-committee, Security, will be much more in depth so as to provide actionable information for the reader of this document.

III.I Legal Sub-committee

The Legal sub-committee was led by a member of the corporation’s Intellectual Property General Counsel Department. She is actively involved in the day-to-day operations of the company with regards to contract negotiation, vendor management and licensing agreements. Leading this sub-committee was a natural fit because she was already involved in reviewing existing licensing agreements that did not make reference to the use of Open Source solutions in the vendors packaged product. This issue, the use of Open Source in a packaged product, is an aspect of Open Source licensing which needs to be carefully dealt with given the increased usage of Open Source solutions as part of the software product vendors packaged solutions. Although this also is an area of concern and needs to be investigated, it was considered out of scope for this effort and hence will not be included in this document.

The research began by going back and reviewing the licenses that were already in place from vendors that did reference Open Source solutions. One of the main aspects of the licensing agreement that needs to be taken into account is the matter of Intellectual Property and who actually has rights to it. Depending on the license agreement, it is possible that a corporation who develops a system to help in the delivery of a service or product that ultimately contributes to the generation of revenues would be required by law under the license to forfeit the revenue or part thereof to the developer of the Open Source software. In addition, the creator of the Open Source Solution may also have legal rights to the system itself and hence the deployment of it on the open market as they see fit.

As with any sort of licensing agreement, there is a lot of grey area that needs to be carefully studied and discussed. Given the variations in licensing agreements, the sub-committee chose to focus their research initially on the

license agreements for the larger Open Source Communities such as Apache as a basis for limiting the scope of this effort. In-depth research into the different licenses was critical for this effort because if not done properly, the corporation could potentially be legally obligated to turnover a system to the Open Source provider that contained information that could help their market competitors or face litigation. Also, it was important to get a sense of the amount of time that might be required to do the individual assessments later on to ensure accurate input into the TCO calculations.

This sub-committee looked at several different licensing agreements that existed in the Open Source community and determined that they varied widely from little or no restriction such as simply crediting the developer of the Open Source to full restriction that granted the developer rights to the end product in which the Open Source software was involved. It is clearly obvious that a for-profit organization in a highly competitive market could not possibly use software which had license restrictions that granted rights to the public at large. The research objective was to determine what level of restriction was acceptable based on the importance and value that the Open Source solution contributed to the development of the corporations system. In addition, the sub-committee needed to determine if it was possible to identify Open Source solutions that delivered the functionality required with licenses that did not expose the corporation to potential litigation.

In some cases, Open Source providers reference generic licenses such as the GNU General Public License (GPL) or Apache Software License. It is much easier for an organization if the Open Source provider uses one of these standard licenses because the research can be a one-time effort rather than a solution-by-solution research effort.

III.II Usage/Deployment Sub-committee

The focus of this sub-committee was to establish a process by which an application development team could submit for review an Open Source solution that they wanted to use in their system. Once approved, the process and system supporting it needed to be capable of deploying new releases, logging users/groups who used the Open Source solution as well as provide a communication mechanism for the corporation to identify and contact users of the approved Open Source Solutions. The leader of this sub-committee was a mid-level technology manager whose normal scope of responsibilities already included oversight of software deployment products.

The first aspect of this effort was to determine whether the fact that the software was an Open Source solution had any bearing on a traditional software deployment process. It was quickly determined that it did not and that the Open Source solution could be treated like any other software product approved for use within the corporation. This greatly simplified the effort because this allowed the sub-committee to use the existing infrastructure and processes for the Open Source solutions.

The sub-committee next began working to determine if the existing deployment tools could accommodate the review, approval, usage tracking and

communication elements. The sub-committee needed to ensure that the existing deployment system(s) could accept requests and route those requests to the appropriate reviewers (Legal, Security, Break/Fix, TCO) based on a pre-defined workflow which the OSSC was developing. Once the solution was approved and made available, the system needed to restrict access to the Open Source solution to only users who had registered and were granted rights to use that Open Source solution.

The previous statement is very important because, should a problem or vulnerability in an approved Open Source solution be identified, the OSSC would need to quickly execute an impact assessment and provide a recommendation on how to address it. The only way to have confidence that the impact assessment was accurate, would be to ensure that all the systems using the solution factored into the assessment. The inventory of systems/users would also act as the starting point for the communication of vulnerability resolutions. Lastly, this restricted access model provides a review/checkpoint for the OSSC to verify the risk level of the system that the Open Source solution would become part of. Dependant on that risk level, the OSSC could determine if that system was too sensitive or critical and therefore refuse the request to use the Open Source solution in that system regardless of the fact that it may already be in use by other systems.

III.III Break/Fix Sub-committee

The Break/Fix sub-committee was charged with determining the best process to identify and provide solutions for bugs and/or vulnerabilities reported in approved Open Source solutions. Since Open Source solutions are not "sold", the typical contractual support obligation established through a purchase agreement does not exist. The sub-committee needed to establish whether or not the Open Source communities themselves provided enough support or if there were third party organizations available to provide support services for the approved Open Source Solutions.

The typical way to receive help from an Open Source community is to be an active participant in that community. This requires a considerable amount of commitment from an individual as well as their employer and does not guarantee that the help will be of any particular level of quality or timeliness. In addition, each Open Source solution could potentially have its own unique community so a corporation would need to have individuals dedicating considerable amounts of time to a community for every Open Source solution approved for use.

An alternative way to have support available for systems using Open Source solutions is to sign support contracts with third party organizations that support the Open Source solutions being used. These support arrangements could consist of anything from "stand-by" support should a problem arise to extensions of the development/maintenance team. As mentioned above, the first problem encountered with this model is that there are very few Open Source solutions being supported by third party organizations and the majority, support only major

products like Linux or Apache Server. This fact leaves a tremendous void in the third party support model that needed to be addressed and understood.

III.IV Total Cost of Ownership(TCO) Sub-committee

The TCO sub-committee's focus was slightly more straight forward than the other three but needed to be sure that it could account for the review and approval processes that the other three sub-committees were still defining. This sub-committee needed to establish a process by which each of the other sub-committees could provide a resource utilization figure as input to the equation so that the TCO could accurately be compared to "Off-The-Shelf(OTS)" software provided by a traditional vendor.

This comparison between the TCO of an Open Source Solution and that of an OTS package can be very difficult because it is not always easy to be sure you are comparing the same things. For example, the vendor of an OTS package may require that you upgrade your software every year to maintain their support agreement. These maintenance/support packages can be very expensive over the life of the software. With Open Source solutions, since there typically is no "official" support from a solution provider you would not be required to perform any upgrades. The obvious drawback to not having contracted support is that if you are using the Open Source solution in a production environment, the development group may be increasingly acting as a support organization rather than a development group. This would not necessarily be the case in an OTS package model. This additional in-house support would have a very large cost associated with it and thus needs to be carefully factored into the TCO equation. The other component which might not normally be factored into the TCO but does have financial implications on the corporation is that these same individuals who are now providing “support” are taking time away from their normal development efforts and could put other project at risk from delivering on time. In addition, if the corporation is not careful or aware of the situation, they may hire additional “development” resources when what they really need is to formalize the support organization and then calculate the costs across projects.

Lastly, there needs to be an acceptable financial cost factored into the TCO equation which accounts for the uncertainty of an Open Source solution. It is more difficult to project the financial impact of a major problem in an Open Source solution than in an OTS package because there is no contract with a vendor that the corporation could leverage. Without a “vendor” under contract, the corporation must incur all costs associated with correcting the problem. This includes not only the resources required to fix the problem but more damaging could be the public relations impact and loss of customer confidence that could lead to lower revenues & sales. It also goes unsaid that the same resources working to fix the problem would place their other responsibilities on hold until the problem was resolved.

III.V Security Sub-committee

As the chairperson for the Security Sub-committee, I decided that I needed to establish a process that could ensure an acceptable level of comfort and risk for each Open Source Solution being submitted for approval. In order to do this, I would first need to clearly understand the differences between in-house developed software and Open Source solutions. I determined that the process would have to provide a picture of who the Open Source developers were as well as the history of the Open Source solution itself. For software developed in-house this is a relatively simple process for the obvious reasons, however, for software developed by an unknown group of individuals, it may not be that simple or worse yet, it may not be that accurate.

The approach that the sub-committee decided to research was whether or not it would be possible to put the "Open Source" solution through a comprehensive Software Quality Analysis(SQA) process. This process would first help to determine whether or not there were any identifiable software vulnerabilities that could potentially be used in the future to compromise the corporation's computer environment. Secondly, the process would include a thorough investigation of the solution provider in order to establish whether or not they were reliable and what level of software quality & support they have delivered in the past with other "Open Source" solutions.

The Software Quality Analysis performed on the "Open Source" solution should be no different than any other traditional SQA effort. As background, a thorough SQA effort will identify structural deficiencies such as unreachable code, unconditional branches into loops, undeclared variables, unused functions, mismatched parameters etc. These types of structural deficiencies may or may not be intentional development errors and there is no way to know for sure whether they can be used by someone in the future as a stepping stone for an attack. For this reason, the SQA effort would evaluate the structural integrity of the Open Source source code and determine if there are any deficiencies. If any are identified, they must be documented and a recommendation as to how they should be addressed would be included in the Source Code Assessment. By eliminating and or documenting these deficiencies the corporation greatly reduces the potential for the Open Source solution to be compromised in the future.

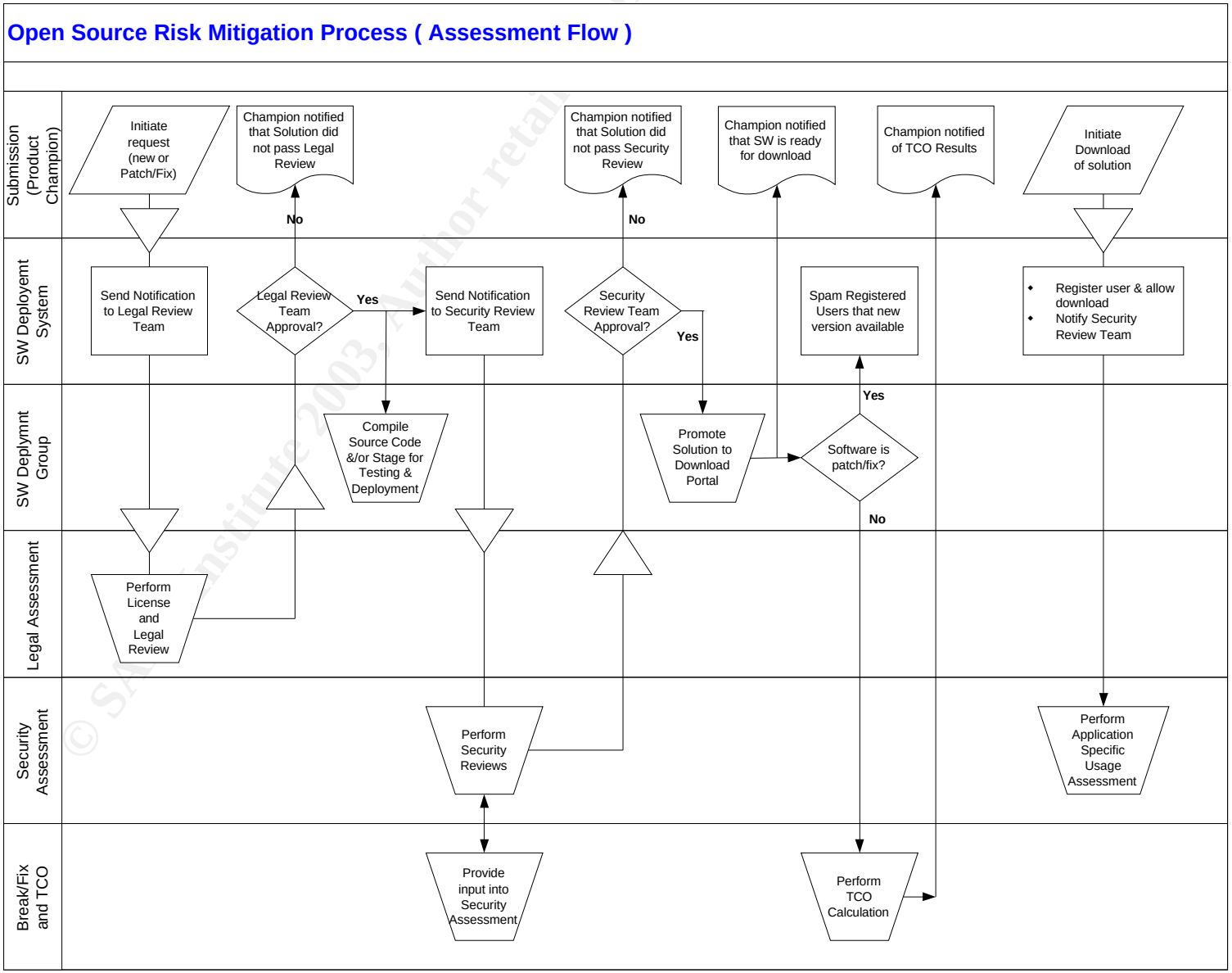
IV The Solution

The solution which resulted from this investigation was a comprehensive but efficient assessment, approval and tracking process for all Open Source Solutions proposed for use within the corporation. A key and primary element before the solution could be implemented was that the ITPEB would be required the modify all IT Policies referencing the use of vendor software to include verbiage addressing how Open Source Solutions could and could not be used. The Open Source Risk Mitigation Process would be referenced in the policies

and all details regarding the process would be documented and posted in the corporate repository for all to access.

The process consists of several assessment points as well as a system facilitating the assessments so that the submitter of the Open Source Solution knows where in the Risk Mitigation Process the request is. The process would begin by requiring that the Application Development person submitting the request be designated as the “Champion” of that Open Source solution. As the “Champion” they would be the primary contact for the review team as well as in the future if other teams decided to use the same Open Source Solution in another application.

The graphic provided below is that of the review and approval workflow for the Risk Mitigation Process. The graphic depicts the different assessment points and teams performing the reviews. Below the graphic is a description of the process and assessment points.



IV.I Request Initiation:

The process begins by having the Champion initiate a request in the Software Deployment System(SWDS) for the review of an Open Source Solution. This review could be for either a new solution or for a patch/fix to a previously approved solution. The review & assessment process does not differ greatly between the two. The only differences between the two are that in a patch/Fix model, a new TCO is not performed and registered consumers of previous versions of the solution are notified of the new patch/fix. In the new solution, a TCO would be performed and there would be no previously registered users to notify. The Product Champion is required to enter the following information into the SWDS.

- Solution Name (i.e. STRUTS)
- Name of Community (i.e. Apache Software Foundation)
- source code of the solution (not a compiled version)
- location of download (i.e. http://www.apache.org)
- Version of source code
- Champion contact details

Once submitted with the required information, the system notifies the Legal Review Team informing them that an Open Source request has been received.

IV.II Legal Review:

The Legal Review Team initiates their review into the licensing and intellectual property of the Open Source Solution. If the Legal Review Team determines that the solution passes their requirements, they update the SWDS with an approved status. The system then sends a notification to the Security Review Team informing them that an Open Source request was received and has been approved by the Legal Review Team. The legal review consists of traditional research into the license agreement for the solution as well as case history if any exists of that license agreement or solution provider. There was no formal process defined for this review. Every legal department has its own process for researching and assessing license agreements and there didn't appear to be any uniqueness about the Open Source research to require a modified process on this case.

If the Legal Review Team however does not approve the solution, the SWDS sends a notification to the Champion informing them of the rejection. At any point during the Risk Mitigation Process, especially if there is a rejection, the Champion can request a formal review meeting with the Open Source Steering Committee members to discuss the solution and present any details to support their case as to why the solution should be re-considered.

IV.III Security Review:

The Security Review Team performs its review by conducting 2 separate assessments in parallel as well as soliciting input from the Break/Fix team as to how exposures and/or bugs could be addressed if the solution is approved.

Some Open Source Solutions can be rather large and as such, the SQA effort of the Source Code Assessment could drastically impact the Total Cost of Ownership calculation. For this reason, the OSSC decided to make the Source Code Assessment optional with the understanding that all recommended solutions must come from only established, well organized and proven Open Source communities. This “optional” decision would be made in conjunction with and with the approval of the OSSC.

This rationale is what essentially led to the establishment of the two separate assessments. The optional assessment would be more technical and address the actual source code of the solution. The other assessment, which is not optional, would address the Solution Provider. A more detailed explanation of each assessment is provided in the paragraphs below.

The goal of the two Security assessments in the "Risk Mitigation Process” is to reduce if not eliminate all exposures that the corporation could inherit by adopting an Open Source Solution. As mentioned above, one is required and one is optional and up to the discretion of the individual requesting the Open Source Solution in conjunction with the various assessment teams. In the early stages of research, it was thought that specific security analysis & testing could be performed on the Open Source Solution to identify potentially malicious code within it. It did not take long to realize that there were no tools available to address that specific need and that there really was no engineering or structural difference between Open Source software and in-house developed software. Manually reviewing every line of code for malicious code is not a cost effective option nor a realistic way of approaching the problem given the complexity of many solutions. The only real difference between Open Source software and in-house developed software is that the developers are not employees or contractors of the organization and therefore have no legal responsibility/accountability to the organization ultimately using the software.

The individual assessments cover both technical and non-technical topics and require research into the community/group that has developed & released the Open Source Solution. Upon completion of the assessments, the corporation should have a very good understanding of potential vulnerabilities, version release processes and the software quality of the Open Source Solution. The results of the assessments will include a “Guidelines For Use” section. This section will provide guidelines for how the Open Source Solution can or cannot be used. A usage recommendation could for example allow an Open Source Solution to be used on internal departmental applications but restricted for use on customer facing Internet applications. Another example might be that the guideline for use prohibits the usage of certain portions of the Open Source Solution but allows others (i.e. "Tag Libraries" which is a component within STRUTS may be rejected while the remainder of STRUTS is approved).

Individuals responsible for executing the two assessments are also responsible for involving the Break/Fix assessment and TCO teams. The Break/Fix team will provide input into the Solution Provider assessment and the TCO team will need details from the Security Assessment team in order to properly compute the projected TCO.

Once the Assessment(s) are completed, the SWDS is updated with an Approved/Rejected status. The SWDS then sends a notification to the Champion informing them of the results of the Security Assessment. Below is a detailed description of the two Security Assessments

IV.III.I Security Review: Source Code Assessment (Optional)

This assessment focuses specifically on the source code of the solution itself, how that source code was written & tested and whether or not any structural flaws exist which could intentionally or unintentionally be used to compromise the corporations' environment. The assessment, which is in the form of questions & answers, provides a guideline of what should be looked for when executing traditional software testing techniques. The assessment contains four question & answer sections and a two part Guidelines for Use section.

The first question & answer section addresses background details of the Open Source Solution such as documentation, the company's experience level with the solution and the age of the current version. The answers to these questions will provide an understanding how mature the solution and group who provided it is with respects to software engineering practices.

The Static Source Code analysis and Dynamic Code analysis sections address the traditional testing techniques that are common practice in the deployment of quality software products. The questions provided in the assessment provide only an outline for investigation based on a traditional analysis. They do not address every minor aspect of a Static or Dynamic code analysis. The individual performing the assessment should use their judgement as to whether deeper investigation is necessary and communicate the results of the more in-depth investigation in an Appendix to the assessment.

The third question & answer section relates specifically to "Security" aspects within the source code. Some of the questions in this section rely on the interpretation of the previous analysis results by the technical expert to help determine if software vulnerabilities identified in the previous analysis could be used to compromise the corporations' computer environment. This section is very important because it begins focusing the individual on the Guidelines for Use section which will be the basis for decision making.

The final section in the assessment is "Guidelines For Use". This section consists of 2 sub topics, which ask the individual performing the analysis to provide their opinion on how the Open Source Solution should be used from both a "Technical Perspective" as well as a "Security Perspective".

IV.III.II Security Review: Solution Provider Assessment (Required)

This assessment focuses exclusively on the solution provider (community/group) and the processes they use related to the development and distribution of their Open Source Solutions. Information gathered from this assessment will be a critical factor in helping the corporation to decide whether or not to approve the solution because it will provide a view into the reliability and maturity of the community/group. The individual executing this assessment may choose to not approve the Open Source Solution for use regardless of the functionality or technical recommendation. That is because, the results of this assessment could indicate a lack of controls in the distribution process, a history of bugs/vulnerabilities previously released or a lack of response to reported bugs/vulnerabilities. The assessment is

composed of five individual sections each targeting specific elements related to the history of the Open Source Solution being reviewed and the provider of that solution.

The first section in the assessment deals with the Criticality, Complexity and Usage of the Open Source Solution. Without clearly identifying the Criticality or Complexity of the solution as well as how that solution would be used in a system, it would be very difficult for anyone to fully understand the level of Risk that the organization will be assuming by approving this solution. The assessment uses a concept called "Overall Criticality Level" which is the product of plotting the Open Source Solution Criticality versus the System Usage Criticality. It is important to do this in the initial assessment so as to reduce or eliminate the need to perform comprehensive assessments of pre-approved Open Source Solutions for every system that wants to utilize the previously approved Open Source Solution. The corporation must first determine an acceptable “Maximum Overall Criticality” level for the use of Open Source Solutions. Then when a solution is assessed, it is assigned an Open Source Solution Criticality value, 1(low) – 5(high). A development group debating whether or not to use an approved Open Solution could then very easily determine if the approved solution would be allowed in their system by checking the product of their System Usage Criticality versus the Open Source Solution Criticality in the matrix. The final decision on usage would be made during the “Application Specific Usage Assessment” which is described below.

Section two addresses the "Provider History" of the community/group that has released the Open Source Solution. It investigates areas such as community size, maturity, support structure and previous success/failures with other Open Source Solutions.

The third section in this assessment tries to identify the current state of the Open Source Solution and how it has been adopted by the IT community at large as well as industry specific adoption by researching the "Solution History". Besides documenting the current and past vulnerabilities it also tries to identify how responsive this community/group has been in the past in addressing reported bugs/vulnerabilities. This section will also look into whether the solution has been adopted by outside organizations or has been included in 3rd party vendor products such as STRUTS, which is currently distributed as part of an IBM packaged solution. Lastly, this section seeks to understand the level of support that might be available directly from the community or from other 3rd party service providers. Input into this section is provided by the Break/Fix assessment team.

The Development & Testing section attempts to identify any special training or tool requirements the corporation will need to take into consideration prior to approving the Open Source Solution. This is important not only from understanding the in-house level of experience with the solution but it also is important to provide these details to the TCO subcommittee so that they can factor it into their analysis.

The last section of this assessment attempts to set proper repair/upgrade expectations by clearly describing the history of "Patch / Version Release" by this community/group. It is important to understand before adoption how, why and when this community provides new releases. A community which releases many versions within a short period of time may possibly not have adopted/developed an acceptable level of rigor in their deployment process. This directly reflects on the software engineering maturity of the community/group. A lack of maturity could cause consumers of this solution to be constantly upgrading their production systems because of vulnerabilities introduced due to poor development and deployment practices.

IV.IV Break/Fix Review/Research:

As mentioned above, the majority of the results of this review/research is delivered as part of the Security Review. This team researches what options are available to the development community should the Open Source Solution encounter bugs or deficiencies. According to many of the Open Source licensing agreements the consumers are required to provide back to the community any

fixes which are developed. Also, should enhancements be developed, those also need to be contributed back to the community for incorporation into the next release. The problem is that there is no guarantee that an enhancement or bug fix will be incorporated into the next release which leaves the consumer in a very difficult situation supporting source code which is not in synch with the community. There is a delicate balance that the consumer must consider for each Open Source Solution to ensure that they remain in synch with the community releases but yet deliver enhancements and/or patches that they developed and utilize in their corporation's system. The statements above all imply that the corporation has authorized the Champion(s) of the Open Source Solution to actively participate in the Open Source communities which may not always be the case. There may be legal as well as security reasons why a corporation may not want to allow their employees to participate in open forums openly for fear that they may unknowingly reveal confidential details about the corporation. Reasons may be as simple as not wanting the Open Source community to know that Company X is using a particular Open Source Solution because should a security exposure be identified, the corporation's Internet based systems could be targeted for attacks.

An alternative to having employees of the company participate in the Open Source community is to establish a service contract with a 3rd party service provider to act as the community interface. In some cases, such as Linux, there are several service delivery vendors which provide this option. In other cases, where the Open Solution is not widely adopted and/or repackaged for sale, there may not be 3rd party service providers. Part of the responsibility of this team is to identify potential service providers and define the best engagement model to support the Open Source Solution if it is approved.

IV.V Total Cost of Ownership:

This portion of the Risk Mitigation Process simply provides data for the Champion and System Owner to act on and does not approve nor reject the request. This team collects information throughout the Risk Mitigation Process from the various assessments teams and compiles it into one TCO figure.

Since the Open Source Solutions do not have any sort of Service Level Agreement associated with them this may cause the system owner to incur additional costs in the areas of, training, support and testing beyond what they might normally spend with a packaged solution from a trusted vendor. In addition, the original requestor/champion, must incur the expense of the full Risk Mitigation Process executed by the Open Source Steering Committee. The time and resources required to execute the comprehensive assessment process alone could justify the use of packaged solutions in some cases. It is critical to not view Open Source Solutions as "free" simply because you can download them from a website without a monetary transaction. It is also as critical to execute the Risk Mitigation Process because a legal infringement or security exposure could cost the corporation significant financial harm far beyond the cost of a purchased solution. Also, since every upgrade and/or version release requires that the Risk Mitigation Process be executed at least in part, the TCO of

an Open Source Solution can quickly grow beyond that of a packaged solution which may not require the same level of testing and research. From a packaged solution perspective however, upgrades typically are not free and can in some cases be very expensive to purchase and/or test.

Lastly, because of the nature of the Open Source model, consumers of an Open Source Solution must stay active in that Open Source community in order to be aware of patches, enhancements, new releases or simply for support purposes. These communities are self governing and typically encourage the consumers of the solutions to contribute back to the community as much information or knowledge as they take. This is the essence of these communities and a big reason why they survive. Open Source Users who simply consume solutions but never give back to the Open Source community can over time become ignored by the active community for lack of contribution. What this ultimately means to the corporation is that the Champion of the Open Source Solution must spend time actively participating in the community in order to stay aware of the progress made with the Open Source Solution as well as establishing credibility should they need to invoke the help of the community to fix a problem. This time spent in the community must be estimated and factored into the TCO equation if it will be performed during the normal working hours.

Upon completion of the TCO calculation the TCO team contacts the Champion and provides them with the TCO results. The Champion along with the Business Owner of the system must then review the TCO calculation and evaluate whether or not the Open Source Solution is a cost effective option or if there are less expensive alternatives which the TCO Team may have provided as a product of their research.

IV.VI Application Specific Usage Assessment

This assessment is performed whenever an approved Open Source Solution is downloaded from the SWDS for use in an application. This could be at the end of the initial Risk Mitigation Process by the Champion or by a totally different team that wants to use a previously approved Open Source Solution. A reason for this assessment is to ensure that all systems utilizing the Open Source Solution do not exceed the Maximum Overall Criticality threshold. This also allows the Security Review team an opportunity to evaluate how the solution will be used in that particular situation. For example, an Open Source Solution might be approved and allowed to be used for an internal Intranet based application. However, the same Open Source Solution might not be allowed to be used in an external Financial related Internet application because the deficiencies identified in the Security assessment may only be exploitable through the Internet and not the Intranet.

This assessment should be relatively brief because it relies primarily on the results from the original Open Source Risk Mitigation Process performed when the Open Source Solution was first requested. The key element in determining whether or not it is allowed is by using the "Criticality Matrix" in Appendix A and plotting the Open Source Solution Criticality identified in the Risk Mitigation

Process versus the system criticality that the development group will determine for their system. As described above, the development group must then decide whether or not the product of the two factors is above the "Maximum Overall Criticality" level. If it is not, then there is typically no need for a more comprehensive review, if it is above the "Maximum Overall Criticality" level, the development group could request that a special assessment be performed taking into account any specific issues they feel might allow the usage of the solution. If the result from the Criticality Matrix exercise is within the Maximum Overall Criticality threshold, the Application Specific Usage Assessment is executed as a validation of results instead of a comprehensive investigation.

V Conclusion

The marketplace is pressuring corporations to aggressively review their expenditures and find less expensive alternatives for their current processes. These pressures can sometimes be so great that a corporation lowers their standards without fully understanding the hazards of their decisions. These hazards are not only expense related but also security and public relations related.

The Open Source Risk Mitigation Process described in the previous pages is a tool for corporations to use when trying to understand why a simple decision to use the “free” Open Source software should be taken very seriously. The ramifications of using just any Open Source Solution without understanding its weaknesses and not implementing compensating controls could be disastrous and much more expensive than a purchased alternative. In some cases, Open Source Solutions fulfill all the requirements with few issues that need to be addressed but that is not always the case. The Open Source Risk Mitigation Process was designed to enable the corporation the ability to use the most cost effective software tools available while identifying, documenting and mitigating the risks associated with the use of those tools.

VI List of references

VI.I General Open Source

"Open Source Gains Mind-Share"
15 March 2003, CIO Magazine | Trendlines
http://www.cio.com/archive/031503/tl_gain.html

Koch, Christopher. "Your Open Source Plan"
15 March 2003, CIO Magazine
http://www.cio.com/archive/031503/opensource.html

Koch, Christopher. "Who are those guys?"
15 March 2003, CIO Magazine
http://www.cio.com/archive/031503/opensource_sidebar_3.html

"Open Source Software - Executive Summaries "
24 April 2003, CIO Magazine
http://www.cio.com/summaries/program/open

Cosgrove Ware, Lorraine. "Open Source Gains Momentum"
03 December 2002, CIO Magazine
http://www2.cio.com/research/surveyreport.cfm?id=48

Cosgrove Ware, Lorraine. "Confidence in Open Source Growing "
07 January 2003, CIO Magazine
http://www2.cio.com/research/surveyreport.cfm?id=51

VI.II Security

Berinato, Scott. "The Open Source (Non-) Debate"
20 June 2002, CIO Magazine
http://www2.cio.com/research/security/edit/a06202002.html

Hurley, Edward. "Open-source security: It's all in the scrutiny"
21 Mar 2002, SearchSecurity.com
http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci811614,00.html

Fisher, Dennis. "Security Fueling Open-Source Adoption"
29 October 2002, eWeek.com
http://www.eweek.com/article2/0,3959,655054,00.asp

Fisher, Dennis. "Open Source: A False Sense of Security?"
30 September 2002, eWeek.com
http://www.eweek.com/article2/0,3959,562220,00.asp

VI.III Total Cost of Ownership

Smith, Tom. "Briefing Book: Open Source's Total Cost Of Ownership"
08 October, TheOpenEnterprise.com
http://www.theopenenterprise.com/story/TOE20021008S0002

"Calculating the Total Cost of Development - Is Open Source Security Really Free?"
01 January 2002, RSA Security - COMDEX White Papers
http://comdex.bitpipe.com/data/detail?id=1027448748_290&type=RES&x=160385877

Schlesinger, Steve. "Open Source Security: Better Protection at a Lower Cost"
March 2003, InfoSecNews.com
http://www.infosecnews.com/opinion/2003/03/19_01.htm

VI.IV Software Quality Assurance

Frequently asked questions about static source code analysis
Cleanscape.net
http://www.cleanscape.net/programming-solutions/code-analysis/lintfaq/index.html

Friesen, Jeff. "Study Guide: Tools of the trade Part 2"
JavaWorld.com
http://www.javaworld.com/javaworld/jw-12-2002/jw-1206-java101guide.html

Mimoso, Michaels. "Basic security tenets apply to open source programs too"
05 February 2003, SearchSecurity.com
http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci878695,00.html

VI.V Legal

Open Source Initiative - Common Public License
Antifork.com - Open Source Initiative
http://www.antifork.org/opensource/licenses/cpl.php

VII Appendix A

Criticality as determined by Source Criticality and Usage Criticality

Open Source Criticality						
(high)5	7	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
(low)1	1	2	3	4	7
	(low)1	2	3	4	(high)5

System Usage Criticality

Overall Criticality Level:

Low	0 - 4
Medium	5 - 12
High	13+

Axis Definition:

Vertical axis - Criticality of the "Open Source" solution
Horizontal axis - Criticality of the system that the "Open Source" solution will be used in

Note:
The values in the Matrix were arbitrary and could be modified to suit a corporations needs.

Last Updated: June 22nd, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Cyber Defence Canberra 2017	Canberra, AU	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MDUS	Jun 26, 2017 - Jul 01, 2017	Live Event
SEC564:Red Team Ops	San Diego, CAUS	Jun 29, 2017 - Jun 30, 2017	Live Event
SANS London July 2017	London, GB	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, JP	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TXUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, SG	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CAUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, DE	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DCUS	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TNUS	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TXUS	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Hyderabad 2017	Hyderabad, IN	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MAUS	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Prague 2017	Prague, CZ	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UTUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NYUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, ILUS	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, AU	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VAUS	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS San Francisco Fall 2017	San Francisco, CAUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FLUS	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NVUS	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, IE	Sep 11, 2017 - Sep 16, 2017	Live Event
SANS Paris 2017	OnlineFR	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced