



Developing a Security Self-Assessment Program Utilizing NIST 800-26 Part II

Welcome back to our discussion on developing a security self-assessment program. We left off last with a discussion on determining the scope of one's self-assessments. We'll conclude in this document by discussing how to develop and deploy an appropriate self-assessment "tool."

Develop self-assessment tools -- Based on the established scope, the means to evaluate compliance with the security requirements (i.e., a tool) must be developed. This tool can be as simple as a word table or as complex as a web-enabled application with an intelligence engine to calculate results and generate reports automatically. There are a number of vendors today who provide such tools at every level of complexity. Regardless of the type of tool developed or used, the following elements should be captured for each self-assessment:

- System name and description
- Contact name and number for individual responsible for the self-assessment for that system
- Each security requirement being evaluated and the following information for each:
 - o The relevant computer security regulation and/or best practice
 - o The method for determining whether the security control is in place or not
 - o The results
- Suggestions from the participants on how to improve the self-assessment program

The organization's audit documentation requirements and how much information has been deemed necessary to adequately evaluate the self-assessment results will drive how much supporting evidence the participants will have to supply with their results.

As previously mentioned, documentation requirements will also have to be determined. In other words, how much supporting evidence will participants have to submit with their assessments? Will supporting evidence be provided only for tests and/or requirements that the participants say they are in compliance with? What type of supporting evidence will be required? If a participant maintains that they have a system security plan, should they submit it with the self-assessment? For specific security settings, will screen shots or appropriate system reports need to be submitted? These considerations should be carefully weighed against 1) what is the necessary amount of supporting evidence to adequately evaluate self-assessment results and 2) the amount of time and resources it will take participants to generate the documentation. It is important that all information collected by the tool during the self-assessment process be safeguarded with appropriate access controls, as it will contain a large amount of sensitive information.

Develop a deployment strategy -- After the primary elements of the self-assessment program have been determined, a deployment strategy should be developed. A deployment strategy will document the elements necessary for implementing the program across the organization. Your deployment strategy should address items including:

Program management -- A program management function for the overall self-assessment program will need to be established. The project manager will be responsible for determining the timing of self-assessment related activities, the budget, staffing, deploying the program and overall quality control. The project manager should

be involved with the program from its inception to ensure a solid knowledge base for its management.

Coordination across the organization -- In order to make the most efficient use of resources, it is important that the lines of communication be established throughout the organization. Effective communication can allow for efficient knowledge transfer, potential cost savings from leveraging previous audit/review findings and functional expertise, a greater level of program acceptance and an overall cohesiveness in program results. Coordination throughout the organization will also help to eliminate duplication of efforts. An example of effective coordination in an organization would be that the internal audit department recently reviewed a system about to undergo a self-assessment and agreed to allow the system owner to utilize the audit results in completing the self-assessment. This would save the system owner time, money and staff resources.

Training and awareness -- Awareness training should be conducted prior to the deployment of the self-assessment program, especially if there is an automated tool involved. Participants will need to be made aware of the process for completing the self-assessments, documentation requirements, how to submit the results and who they can contact with questions. It would be beneficial to provide the participants with training materials that they can take with them as points of reference. Additionally, training provides a good opportunity to discuss the benefits of the self-assessment program in order to gain support and buy-in from the users. At the conclusion of training, participants should be given an idea of what to expect as the next steps in the self-assessment process. This should include: when they will be given access to the materials, when the self-assessments should be submitted and what the reporting process is going to be.

Timing -- How valuable would a self-assessment be if it took participants three years and a total of \$100 million to complete? The organization should establish realistic timeframes for completing self-assessments. If a pilot was conducted for the program, it should provide a good basis for estimating the amount of time required to complete a self-assessment. In order to get comprehensive coverage of all critical systems and their components (operating systems, applications, network components, etc.), a rotation schedule will likely need to be developed for completing self-assessments. An example of a rotation schedule is outlined below:

- Year 1: All NT and Unix servers housing critical applications
- Year 2: Application and database reviews for the most critical applications
- Year 3: A selection of firewalls and routers connecting the components of critical systems

The rotation schedule should be developed based on the self-assessment program's objectives, applicable requirements/best practices, resource requirements and ensuring that critical systems receive adequate coverage.

Help desk function -- A help desk function should be established to assist participants with completing the self-assessment. This will help serve a number of purposes, including ensuring that the assessment results are of an acceptable quality, submitted timely and complete. It will also help to identify flaws in the self-assessment process and/or tools. Evidence of flaws would be seen as an increase in the occurrence of calls concerning a particular question or aspect of the self-assessment program. Individuals staffing the help desk should have an in-depth knowledge of the self-assessment process and be available during the timeframe established for the self-assessments.

Data collection -- A determination will need to be made as to how participants will be required to submit self-assessment results. Examples of options for submittal include hardcopy reports or electronic files (via email, storage media or an online application).

This determination should be based on a number of factors including the level of effort and cost to submit, maintain and secure the information.

Reporting -- A process will need to be developed for evaluating the self-assessment results and drafting reports of findings to return to the participants. At minimum, the following elements should be included in the report:

- Findings: Documents the identified vulnerability
- Requirements: Documents the applicable computer security requirement or best practice not being met
- Risks: Describes the risk of not mitigating the vulnerability
- Recommendations: Provides a suggestion to mitigate the vulnerability


Additionally, overall program results should be compiled and reported. If a metrics program were being utilized, a baseline would need to be established during the first implementation of the self-assessment program for future comparison.

Retention requirements -- Retention requirements should be established for the self-assessment results. A number of factors will influence how long an organization decides to maintain the documentation. These potentially include legal requirements, internal policies, the rotation schedule (e.g. would documentation be maintained, for reference purposes, until that particular self-assessment was conducted again) and the cost of maintaining the information.

Lessons learned -- Throughout the self-assessment process, lessons learned should be documented and, at predetermined intervals, compiled for integration into the program. This will ensure continual improvement of the program.

Change management -- A change management strategy will need to be maintained for updating both the tool and the overall program elements. This strategy should be in compliance with established organizational policies.

In closing, properly developed and implemented self-assessment program has the potential to provide your organization with a valuable assurance tool for evaluating the state of your network security programs. We hope that you have found this information useful, and that you will put this plan into action (if appropriate) to ensure the security and compliancy of your network.



References

Fredholm, William. "Web Application Security – Layers of Protection". SANS Infosec Reading Room. 26 January 2003. URL: <http://www.sans.org/rr/papers/index.php?id=965>.

National Institute of Standards and Technology (NIST). "NIST Special Publication 800-26: Security Self-Assessment Guide for Information Technology Systems." NIST Special Publications. November 2001. URL: <http://csrc.nist.gov/publications/nistpubs/800-26/sp800-26.pdf>.

National Institute of Standards and Technology. "NIST Special Publication 800-55: Security Metrics Guide for Information Technology Systems." National Institute for Standards and Technology Special Publications. July 2003. URL: <http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf>.

