Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Using a Capability Maturity Model to Derive Security Requirements

A security engineer is often assigned to a project that already has defined security objectives. But on occasion, the security engineer may be tasked with the initial definition of the objectives. While this assignment may be exciting because of the important role the security engineer is to play, it may also be somewhat daunting due to the large solution space. In order to guide one's efforts in this task, the security engineer could turn to the Systems Security Engineering Capability Maturity Model (SSE-CMM). This mo...

Copyright SANS Institute
Author Retains Full Rights

AD

Using a Capability Maturity Model to Derive Security Requirements

GSEC Practical v1.4b Option 1
Mike Phillips
March 13, 2003

Abstract

A security engineer is often assigned to a project that already has defined security objectives. But on occasion, the security engineer may be tasked with the initial definition of the objectives. While this assignment may be exciting because of the important role the security engineer is to play, it may also be somewhat daunting due to the large solution space.

In order to guide one's efforts in this task, the security engineer could turn to the Systems Security Engineering Capability Maturity Model (SSE-CMM). This model provides industry best practice guidance without being specific as to how security solutions are implemented. The SSE-CMM provides a broad list of "base practices" from which the security engineer can benefit when defining the objectives of the security implementation. This paper will discuss the use of these base practices in the formation of security requirements.

The Problem of Deriving Requirements

Defining the software engineering requirements for a large computing project is never an easy task, and without a clear focus, it is easy to get lost in a mountain of changing customer needs, fluctuating budget targets, and surprisingly mobile schedule milestones. Attempts to define security engineering requirements are often met with the same pitfalls. However, the need for software requirements is more widely recognized than the need for security requirements. Therefore, the tools and processes used for generating software requirements are more mature than those used for generating security requirements.

When one attempts to produce a complete list of security requirements for a large computing project, it becomes apparent that there is not much information available in the way of standardized requirements. In fact, unlike with software development, it is likely that the customer is not well informed about the security objectives that are desired. The customer may not know what security steps are necessary, prudent, or sufficient. The customer just knows that the system needs to be secured. It is up to the security engineer to determine what steps should be taken, explain the necessity of these steps to the customer, and convince the customer that the defined steps are enough to secure the system.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

With that much latitude, the security engineer may begin to believe that the security problem is unbounded, and in a sense, it is. But even though it may not be possible to completely eliminate all vulnerabilities, it is possible to minimize or manage vulnerabilities in a way that accomplishes realistic goals (SANS, Schultz).

The security engineer must use experience and good judgment to generate the proper requirements within the constraints that are given. Consider these situations:

- The Finance department is developing a new accounting system with a web-based interface. They want the system to be secured, but since the system is not a direct producer of income for the company, funds are limited. The security engineer must provide clear levels of security with an estimate of cost for each so that the Finance department can perform the necessary cost-benefit analysis.
- An external customer wishes to develop a large software system that will tie all its sites and functions together. The security engineer must show the customer a complete solution that does everything possible to avoid an embarrassing compromise.
- A customer realizes that a computing system that was developed years ago has few modern security features. The security engineer must arrive at a solution that will secure the system without compromising existing functionality or causing undue down time during implementation.

When faced with any of these situations, the security engineer's most basic need is for a logical approach to use when addressing the problem. This approach must cover all the bases, have a defined endpoint, and provide proof of security to the customer.

An approach that covers all the bases must be flexible. No one list of security requirements will work for all projects. So the approach should be one that emphasizes a way to generate appropriate requirements without specifying what those requirements are. The approach must cover the entire life cycle of the project, rather than just the development phase, or just the operations phase.

The chosen approach must have a defined endpoint, because the system must eventually become operational. The approach must use a defined set of areas to investigate in the search for requirements. Those areas should be clearly explained in order to focus the investigation on the desired questions.

A good approach to generating requirements will support the security engineer's efforts to convince the customer that the information in the system is confidential, trustworthy, and available. The approach itself should be a selling point; it should be a reasoned, logical approach that instills confidence in the customer that the security engineer can arrive at the right solution. The approach should also

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

generate data that can be used to reassure the customer. The requirements themselves, when reviewed with the customer, should provide confidence, as will security planning documents for which the requirements call.

With a logical approach identified, the security engineer can begin to develop requirements with a reasonable expectation of eventual success.

Beyond having a logical approach, the security engineer also needs a plan for future events such as changes in both funding and threats. The act of generating requirements should provide insight into possible areas of improvement that are not currently funded. This information can be used to argue for greater funding or to take advantage of an unexpected budget surplus. Also, the process of generating requirements should be defined and repeatable so that emerging threats can be dealt with efficiently and with confidence that the security posture that was achieved in the past can be maintained or achieved again. With a good approach to requirements generation, the security engineer can lay the groundwork for handling these future events gracefully and effectively.

Admittedly, that is a lot to ask of a process for generating security requirements, but it is similar to the needs of the software requirements generation process. Many in the software industry have adopted the Capability Maturity Model (CMM) approach to defining processes for software development. A CMM goes well beyond simply generating requirements, but it might provide a useful starting point for the current dilemma of ferreting out security requirements from an overly vague Statement of Work document.

The Capability Maturity Model

A Capability Maturity Model (CMM) is “a model for judging the maturity of the … processes of an organization and for identifying the key practices that are required to increase the maturity of these processes (CMSEI, CMM).” The idea behind a CMM is to define areas of a project that should have processes associated with them (“process areas”) and then to measure the application of those processes (“capability level”) in an organization. A more “mature” organization is defined as one whose processes are better defined and managed. Such an organization is said to have a higher capability level than a less mature organization. Note that the presence of processes does not guarantee that the outcome of a project will be successful. But the presence of processes and the adherence to them by the organization should provide some insight into the ability of the organization to accurately predict the outcome and to repeat success achieved on earlier projects.

The Software Engineering Institute (SEI) at Carnegie Mellon University is a leading creator of CMMs. Three examples from the SEI are the Software CMM (SW-CMM), the Systems Engineering CMM (SE-CMM), and the CMM Integration

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

(CMMI). The SW-CMM and SE-CMM apply to specific areas of the computer system development realm. The CMMI is an effort to provide a single model that is integrated across disciplines, such as software and systems engineering (CMSEI, CMMI 2.2.4).

The International Systems Security Engineering Association (ISSEA) has also developed a CMM, the Systems Security Engineering CMM (SSE-CMM). While the International Organization for Standardization (ISO) has accepted the SSE-CMM (ISSEA, Press Release), the SSE-CMM is certainly not as well known as its SEI counterparts, just as the field of security engineering is not as widely practiced as software or systems engineering. It remains to be seen whether or not the SSE-CMM ever becomes a widely used approach. But the concepts it espouses provide a useful foundation for the generation of security requirements, even if its deeper applications are not explored.

The SSE-CMM defines five capability levels (SSE-CMM, p. 61):

- Level 1 – Base practices are performed informally
- Level 2 – Base practices are planned and tracked
- Level 3 – Base practices are well defined
- Level 4 – Base practices are quantitatively controlled
- Level 5 – Base practices are continuously improving

The SSE-CMM defines eleven security-related process areas. They are defined in alphabetical order to avoid implications of a sequence (SSE-CMM, p. 37):

- PA01 – Administer Security Controls
- PA02 – Assess Impact
- PA03 – Assess Security Risk
- PA04 – Assess Threat
- PA05 – Assess Vulnerability
- PA06 – Build Assurance Argument
- PA07 – Coordinate Security
- PA08 – Monitor Security Posture
- PA09 – Provide Security Input
- PA10 – Specify Security Needs
- PA11 – Verify and Validate Security

Using the SSE-CMM to Drive Requirements Generation

These process areas, when measured against the capability levels, can be used in a variety of ways to assess the maturity of an organization's abilities. For instance, a security engineering organization can perform a self-evaluation of its efforts with the intent of identifying weak spots and making improvements. A customer could use this type of evaluation to help determine the fitness of a potential supplier for the work the customer wishes to perform. Or a project might

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

provide an external evaluation of this type as input to the customer to build confidence in the adequate security of a developed system based on the assessed capabilities of the security engineering organization (ISSEA, SSE-CMM).

But the use of the SSE-CMM that is applicable to this forum is that of driving the generation of appropriate security requirements. For each process area, one or more goals are defined, as well as a number of base practices. These base practices detail activities that a security engineering organization should undertake during the development of a computing system. All projects are different, so not every base practice will apply in all situations, but these base practices represent security engineering best practices, so they are all worth considering when developing requirements.

Technology alone cannot ensure the security of a system. In fact, the "most advanced equipment and security safeguards are to no avail if all the users are not properly trained to be part of the security plan (Fulton, Lessons learned)." This principle means that most good security approaches will include the definition of certain operational procedures to ensure that the system is maintained in a secure state long after the development phase is over. Many of the base practices lend themselves to requirements that deal with the development of various security-related procedures and policies rather than dealing strictly with the development of technical solutions.

Every security engineer will approach the base practices with different training and experiences, so requirements will vary. This paper will attempt to provide a few examples of high-level requirements that might be generated for a fictional Unix-based computing system based on the base practices associated with the different process areas. These examples are not meant to be complete or even self-consistent across process areas; that type of exhaustive analysis is beyond the scope of this paper. For clarity, the process areas will be dealt with in four groups:

- Architecture design (PA07, PA09, PA10)
- Security assessment (PA02, PA03, PA04, PA05)
- Operations and maintenance (PA01, PA08)
- Convincing the customer (PA06, PA11)

For each area, the base practices as defined in the SSE-CMM will be listed (e.g., BP.07.01 is the first base practice of PA07) (SSE-CMM, p. 315-318).

Example Requirements Derived from Base Practices

Architecture Design (PA07, PA09, PA10)

BP.07.01	Define security engineering coordination objectives and relationships.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

BP.07.02	Identify coordination mechanisms for security engineering.
BP.07.03	Facilitate security engineering coordination.
BP.07.04	Use the identified mechanisms to coordinate decisions and recommendations related to security.

The goal of PA07 is to involve all the relevant parties in security-related decisions and to properly disseminate all decisions to those who play a part in approving, implementing, or verifying those decisions. One of the parties that should be involved in decision-making at some level is the customer. Eventually, the customer must be convinced of the security of the system, so involvement in the decision making process is important to future success.

With PA07's emphasis on communication, one would expect the related requirements to deal heavily with defining processes. Some example requirements that might be helpful are as follows.

1) Create a working group to address security issues. Establish a monthly meeting schedule and include representatives from the following organizations: systems engineering, software development, security engineering, and the customer. Issues that might be addressed include the establishment of security policies for passwords, a decision on protocols to be allowed to pass through the firewall, a directive regarding the use of dial-up connections, and a review of the current plans for the deployment of LDAP.
2) Create an "Ask Security" internal web page that is accessible to everyone on the project to encourage open communication with the security engineering group. Distribute weekly security tips that explain various security restrictions of the system being developed and their effect on functionality.
3) Establish a security review board that approves all security-related decisions. This board would have the responsibility for creating and maintaining the security baseline for the project. It would work closely with the customer to ensure that budget and schedule restrictions are observed when reviewing a proposed change to the baseline.

BP.09.01	Work with designers, developers, and users to ensure that appropriate parties have a common understanding of security input needs.
BP.09.02	Determine the security constraints and considerations needed to make informed engineering choices.
BP.09.03	Identify alternative solutions to security related engineering problems.
BP.09.04	Analyze and prioritize engineering alternatives using security constraints and considerations.
BP.09.05	Provide security related guidance to the other engineering groups.
BP.09.06	Provide security related guidance to operational system users and administrators.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The goal of PA09 is to drive security decisions into the development of the system and to provide the best security engineering solution available. Once the security engineering organization makes good decisions regarding the proper development of the system to ensure security, it is necessary to support the organizations that are developing the system so that those decisions are properly and consistently implemented.

The requirements associated with PA09 should be focused on translating security decisions into a working part of the system. Here are some example requirements.

1) Perform trade studies to select the most appropriate products for the following areas: firewalls, switches and routers, web servers, and directory services. Trade studies should take into account such criteria as ease of installation, ease of configuration, quality of customer support, robustness of feature set, reliability, industry ratings, and cost. Criteria should be weighted appropriately. The security review board should approve the results.
2) Develop a checklist of common coding mistakes that can lead to security vulnerabilities. Circulate this list in the software development organization, and select a representative sample of newly developed code to review against the checklist.
3) Inform development organizations of default or common capabilities that will not be available due to security measures. For example, reducing dependence on NFS might increase the replication of data, which might lead to a need to purchase more hard disks. Or the elimination of the r-commands might necessitate redesigning a script that initiates processes on multiple hosts. Informing other organizations of these restrictions earlier rather than later will reduce cost and schedule impacts.
4) Establish a working list of potential collisions between desired system functionality and the expected security implementation. For instance, an application may expect to allow users to create new database accounts instantly, which conflicts with a security implementation that requires human approval for new accounts. For collisions that require more than a cursory explanation, formalize the explanation and decision process by presenting the issue to the security working group or the security review board. Prepare a cost-benefit analysis that compares multiple solutions involving changes in both system functionality and security implementation.
5) Prepare and maintain a list of the ten highest priority unresolved security issues. Assign realistic due dates to each item, and present the status of the list at the working group meetings.
6) Security engineering must attend design reviews held by other engineering organizations. At these meetings, security engineering can answer questions from reviewers about how the design fits with security and identify security issues that the other organizations might not have considered.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

7) Provide a security plan that includes operational procedures for system maintenance as well as system operation. Maintenance procedures might include backups, restorations, account creation, firewall maintenance, and router access control list modifications.

BP.10.01	Gain an understanding of the customer's security needs.
BP.10.02	Identify the laws, policies, standards, external influences and constraints that govern the system.
BP.10.03	Identify the purpose of the system in order to determine the security context.
BP.10.04	Capture a high-level security oriented view of the system operation.
BP.10.05	Capture high-level goals that define the security of the system.
BP.10.06	Define a consistent set of statements which define the protection to be implemented in the system.
BP.10.07	Obtain agreement that the specified security meets the customer's needs.

The goal of PA10 is to document the overriding security goals of the system and to gain agreement from the customer on those goals. This documentation is the translation of the customer's view of the system and how it should operate as a part of the whole business into a finite set of achievable security goals. These goals are somewhat abstract and will eventually be broken down into more specific high-level requirements. Those requirements will then be further detailed into derived requirements that specifically state how to implement security mechanisms for the project.

PA10 should generate requirements that specify the customer interaction and the documentation to be expected during the establishment of security goals. Some example requirements are as follows.

1) Create a list of security goals that are agreed upon by the customer and the security organization. These goals might include such things as the establishment of a de-militarized zone (DMZ), the exposure of a web server to the Internet, the use of trusted or hardened operating systems in systems that are likely targets of attack, the support of the File Transfer Protocol (FTP) through a firewall, the avoidance of wireless networking, the use of tape backups, and the implementation of redundant firewalls and routers to achieve high availability. These security goals should be extensive enough to convince the customer that the important data is secured, but not so sweeping as to make the cost unreasonable. The security goals should also be considered by the customer in light of the functionality of the system for the eventual users. The goals must not require an implementation that prevents the users from accomplishing reasonable tasks that are in line with the plans of the customer.
2) Develop a prototype system that implements the security goals in a limited fashion. Demonstrate the prototype to the customer to obtain approval for the functionality of the system as well as its security.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

3) Provide documentation of the legal implications of the system. These implications might include security for financial transactions, confidentiality for medical records, disclaimers attached to information given to users, forms regarding consent to monitoring by system administrators, privacy policies for data that is collected, and liabilities incurred by unexpected downtime or breaches in security.
4) Create a security view for the system to be developed. This view should document all aspects of the system that have to do with security, including areas such as the high-level security requirements, network features that support security, the operational concept for adding new users, and what services will be available on which hosts.
5) Document the security policies to be implemented in the system. Review the list to ensure self-consistency as well as consistency with the defined security goals. These policies might include disabling trusted hosts, requiring the use of SSL where possible, expiring user passwords after 60 days, installing and maintaining mail filters that check for viruses, performing nightly backups, and banning the downloading or installation of unauthorized software.
6) Create a disaster recovery plan that deals with areas such as storage of data backups, emergency contacts, alternate facilities to be used, financial mechanisms needed to fund recovery, and a prioritized list of capabilities to restore.

Security Assessment (PA02, PA03, PA04, PA05)

BP.2.01	Identify, analyze, and prioritize operational, business, or mission capabilities leveraged by the system.
BP.2.02	Identify and characterize the system assets that support the key operational capabilities or the security objectives of the system.
BP.2.03	Select the impact metric to be used for this assessment.
BP.2.04	Identify the relationship between the selected metrics for this assessment and metric conversion factors if required.
BP.2.05	Identify and characterize impacts.
BP.2.06	Monitor ongoing changes in the impacts.

The goal of PA02 is to document the impact of the various threats to the system. The type of impact should be identified as well as a quantitative value of the impact. This characterization will be used in the ensuing determination and prioritization of risks.

PA02 should generate requirements that direct the identification and quantification of impacts. Some example requirements follow.

1) Document potential impacts to the system based on previously identified threats. If a threat was to occur and the system was vulnerable to that threat, identify the nature of the impact. Impacts can be felt in such areas

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

as financial stability, legal liability, public relations, competitive advantage, possession of trade secrets, system capabilities, and harm to personnel.

2) Quantify each impact. Ways to measure impacts include days of lost schedule, dollars of lost revenue, number of lost customers, and number of severe injuries. It is difficult to compare the different units, so define a mapping to a common unit such as low/medium/high. For instance, an impact of more than one million dollars may be defined as high, as might an impact of more than 20 lost customers. In that case, those two impacts would be considered equivalent.
3) Establish a regular review of impact information to account for changes in the quantification, for the realization of new impacts, and for the elimination of existing impacts.

BP.3.01	Select the methods, techniques, and criteria by which security risks, for the system in a defined environment are analyzed, assessed, and compared.
BP.3.02	Identify threat/vulnerability/impact triples (exposures).
BP.3.03	Assess the risk associated with the occurrence of an exposure.
BP.3.04	Assess the total uncertainty associated with the risk for the exposure.
BP.3.05	Order risks by priority.
BP.3.06	Monitor ongoing changes in the risk spectrum and changes to their characteristics.

The goal of PA03 is to combine the threats, vulnerabilities, and impacts into identifiable risks. After identifying the risks, they must be prioritized and assessed for levels of uncertainty. The aim of this activity is to ensure that the most important risks are addressed first. The most important risks may not be resolved first due to time, money, or other factors, but they should be addressed and consciously set aside before less important risks are resolved.

PA03 should generate requirements that direct the identification and prioritization of risks, as well as a mechanism to ensure the risks are addressed. Some example requirements include the following.

1) Define the controlled environment in which risks will be examined. There are an unlimited number of risks that face every system. Many of these risks are infrequent, unlikely to cause damage, and not very harmful when they do cause damage. A scope must be established for the examination of risks that includes boundaries on the risks to be examined and assumptions about the operating conditions for the system. For instance, the assumption that nobody will ever install an unauthorized modem may be valid in an environment that has strict physical and personnel controls.
2) Compare the previously defined threats and vulnerabilities to one another to locate likely areas for problems. If a threat occurs frequently and the system is specifically vulnerable to that threat, then the impact of the threat must be examined to ascertain the level of exposure. If a threat is infrequent or the system is not vulnerable, the impact of the threat is less

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

important. For instance, when viewing equipment failure as a threat to availability, the impact of the threat is of greater importance if the system does not have redundant hardware, thereby increasing its vulnerability.

3) Using the threat/vulnerability/impact comparisons, prioritize the risks. A high threat, high vulnerability, high impact risk should also have a high priority. If any of those areas is low, the risk has a lower priority. And if all three areas are low, the risk is generally unimportant, compared to the other risks.
4) Define the uncertainty level associated with each risk. If the risk is difficult to quantify, then the uncertainty level will be correspondingly high, which may affect the prioritized ranking of the risks. For instance, if a fire would cause the repair or replacement of a system, the impact may be the total replacement cost. But the possibility of a less costly repair introduces uncertainty that might lower the risk somewhat.
5) Assign the responsibility of risk management to the security review board. The board should assess the prioritization of risks, decide which risks should be addressed, and assign due dates and responsible individuals for the resolution of risks. They should also establish a regular review of risk information to account for changes in threats, vulnerabilities, and impacts.

BP.4.01	Identify applicable threats arising from a natural source.
BP.4.02	Identify applicable threats arising from man-made sources, either accidental or deliberate.
BP.4.03	Identify appropriate units of measure, and applicable ranges, in a specified environment.
BP.4.04	Assess capability and motivation of threat agent for threats arising from man-made sources.
BP.4.05	Assess the likelihood of an occurrence of a threat event.
BP.4.06	Monitor ongoing changes in the threat spectrum and changes to their characteristics.

The goal of PA04 is to document the threats that could occur with regard to the system. These threats could be man-made or natural; they could be deliberate, accidental, or random. After defining the threats, they must be characterized regarding the likelihood of their occurrence. Threats that are unlikely to occur are less important than threats that will almost certainly occur. This characterization will be used in the ensuing determination of risk.

PA04 should generate requirements that direct the identification and measurement of threats. Some example requirements follow.

1) Document potential threats against the system. Include such items as malicious software; operator error; hardware or infrastructure failure; severe weather; fire; cyber-attacks such as denial of service, theft of data, and data destruction; and non-computer criminal activity such as vandalism and theft.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

2) Rate the likelihood of each potential threat. Rate threats within categories and across categories. For instance, tornadoes are much more likely in Oklahoma than hurricanes, but neither one is as likely as a power outage or malicious software. To facilitate comparison, assign an approximate frequency of occurrence for each threat.
3) When rating the likelihood of man-made threats, take into account the motivation and capability of the threat agents. High-profile or controversial entities are more likely to face malicious man-made threats than low-profile entities. Entities with valuable data are more likely to face the threat of theft than entities without valuable data.
4) Establish a regular review of threat information to account for changes in frequency, severity, and nature, as well as the introduction of new threats and the obsolescence of old threats.

BP.5.01	Select the methods, techniques, and criteria by which security system vulnerabilities in a defined environment are identified and characterized.
BP.5.02	Identify system security vulnerabilities.
BP.5.03	Gather data related to the properties of the vulnerabilities.
BP.5.04	Assess the system vulnerability and aggregate vulnerabilities that result from specific vulnerabilities and combinations of specific vulnerabilities.
BP.5.05	Monitor ongoing changes in the applicable vulnerabilities and changes to their characteristics.

The goal of PA05 is to document the vulnerabilities that could exist within the system as it is currently defined. These vulnerabilities are weaknesses in the system that could be exploited if the right circumstances were to occur. This list of vulnerabilities will be used in the ensuing determination of risk.

PA05 should generate requirements that direct the identification of vulnerabilities. Some example requirements follow.

1) Document potential vulnerabilities in the system. Include such items as the presence of a wireless network, connections to the Internet, incoming mail, easily removable hard drives, the capability for users to access the system without identifying themselves, single points of failure in the system, limited bandwidth in external connections, the existence of open ports on exposed machines, and the lack of physical security measures surrounding the system.
2) Document the interaction of different vulnerabilities. For instance, easily removable hard drives and a lack of physical or personnel security measures might lead to a vulnerability like theft of data. Connections to the Internet combined with an out-of-date operating system might lead to a vulnerability to easily obtained root access.
3) Perform periodic vulnerability assessments with Nessus. Eliminate false positives and correct all security holes. Analyze all security warnings and

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

security notes to determine if corrective action is warranted. Report the results of the assessment to the security working group (Cole, p. 136, 139).

4) Establish a regular review of vulnerability information to document new vulnerabilities that are introduced and vulnerabilities that have been eliminated, as well as changes in existing vulnerabilities.

Operations and Maintenance (PA01, PA08)

BP.1.01	Establish responsibilities and accountability for security controls and communicate them to everyone in the organization.
BP.1.02	Manage the configuration of system security controls.
BP.1.03	Manage security awareness, training, and education programs for all users and administrators.
BP.1.04	Manage periodic maintenance and administration of security services and control mechanisms.

The goal of PA01 is to make sure that planned and intended security features are actually used in practice to keep security at the intended level throughout the lifetime of the system. Various changes have the potential to affect security, such as:

- New employees – as less-experienced employees are hired to operate the system and more-experienced employees are reassigned, important knowledge about maintaining security can be lost. It is important for this information to be captured and communicated.
- Hardware and software upgrades – each new release may carry its own special security issues, so there should be a clear plan for securely incorporating changes in technology.
- Facility changes – Changing the physical location of the tape backup area could result in an increased exposure to fire, theft, or simple misplacement. These types of changes must be reviewed from a security standpoint, even though the connection between physical facilities and computer system security is not always obvious.

Because PA01 is concerned with ongoing maintenance issues, the creation of procedures to support security is a focus of the related requirements. Here are some examples.

1) Define the individual roles involved in security during the operations and maintenance phase; specify the training that the roles need and the authority that the roles have. Examples of roles might include:
    a. Security officer – the responsible person, makes final decisions, reviews security incidents, trained in the nature of threats
    b. Security architect – head technical person for security, makes technical decisions to implement policies provided by the security officer, trained in the nature of threats and how they apply to the technology in use

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

c. Administrator – upgrades system, configures system, full access requires complete trust from security officer, reviews security events in search of incidents, trained in system administration and the security features of various products
d. Operator – runs system according to rules, reports security events, trained to operate the system
e. User – uses system according to rules, does not share access with others, trained to use the system
f. Developer – aware of security guidelines when creating fixes or enhancements that may or may not be security related, raises issues to security architect, trained to recognize and eliminate specific security holes at the lowest level
g. Tester – tests security changes, tests non-security changes, always on the lookout for potential security issues, provides the last line of defense before a change goes into an operational system, trained in the proper behavior and security approach of the system

2) Create a security plan for system configuration and maintenance. The plan might provide information on such things as:
   a. Handling application updates from the software configuration management system and patches from vendors
   b. Defining ownership and permissions for configuration data associated with firewalls, routers, and servers
   c. Comparing system configuration to related interface control documents to verify IP addresses, protocols, and allowable traffic
   d. Documentation necessary to track configuration changes, such as date of change, person making change, reason, problems found during and after implementation
   e. History of security requirements modifications, with each modification traced back to its approval authority and information on the method of implementation
   f. Steps necessary to sanitize the system to eliminate potentially compromising test data before the system becomes operational for the first time

BP.8.01	Analyze event records to determine the cause of an event, how it proceeded, and likely future events.
BP.8.02	Monitor changes in threats, vulnerabilities, impacts, risks, and the environment.
BP.8.03	Identify security relevant incidents.
BP.8.04	Monitor the performance and functional effectiveness of security safeguards.
BP.8.05	Review the security posture of the system to identify necessary changes.
BP.8.06	Manage the response to security relevant incidents.
BP.8.07	Ensure that the artifacts related to security monitoring are suitably protected.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The goal of PA08 is to identify breaches, attempted breaches, and possible future breaches of the security mechanisms, and to ensure that the proper steps are taken in those situations. Ideally, the proper application of the defined procedures will prevent the introduction of new vulnerabilities. PA08 is concerned with both internal and external security incidents.

Like PA01, PA08 is concerned with the ongoing operation of the system rather than development, so the creation of procedures to support security should be reflected in the requirements. Here are some example requirements that might be useful.

1) Create a log for recording significant security events. Not all events should be recorded (e.g., not every successful login), but only those events deemed worth recording (e.g., a successful login following numerous failed attempts). This log does not show everything known about the event, but it contains log references and other information that will support investigation of the event at a later date if such an investigation is needed.
2) Create a security plan for event monitoring that deals with system configuration changes, facility interruptions, abnormal application behavior, and examples of noteworthy log entries for such items as routers, firewalls, syslog, sulog, loginlog, and security-specific application logs. This plan should also establish criteria for notification of the security officer and an outline for the handling of security incidents.
3) Create a security plan for the reevaluation of threats, vulnerabilities, impacts, and risks. This reevaluation should be at least periodic, with potential unscheduled reevaluations in the case of certain events. This plan should define the level of formality, the responsible individual, and the role of the customer in the process. This plan might include periodic testing of security readiness with the use of password cracking tools, network scanning tools, and comparisons of the planned system configuration to the actual configuration.
4) Define the security metrics to be collected during normal operation of the system, such as the number of successful and failed logins, the number and severity of security incidents, the number of security policy violations, and the number of weak passwords in use (Lowans, p. 2). Each metric should include a threshold for variance from the baseline that triggers notification of the security officer.
5) Protect security monitoring artifacts from unauthorized modification or deletion. Update the permissions on sulog, loginlog, and syslog so that only the system administrator has access (Aton, p. 6). Migrate logs to an automatic tape storage device daily. Restrict physical access to the tape storage area to authorized personnel only.

Convincing the Customer (PA06, PA11)

BP.6.01	Identify the security assurance objectives.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

BP.6.02	Define a security assurance strategy to address all assurance objectives.
BP.6.03	Identify and control security assurance evidence.
BP.6.04	Perform analysis of security assurance evidence.
BP.6.05	Provide a security assurance argument that demonstrates the customer's security needs are met.

The goal of PA06 is to convince the customer that the system being produced is actually secure. The previously established security goals must be addressed with documentation that shows how those goals are being met. This documentation should include proof that the security requirements derived from the goals are being met in such a way that the system remains operationally useful.

PA06 should generate requirements that specify the customer's expectations for a convincing case showing the security of the system. Here are some example requirements.

1) Generate a requirements verification traceability matrix. This matrix should map the customer-approved security goals to high-level requirements, to low-level derived requirements, and to test cases performed to verify the requirements. The customer should be involved in approving the high and low-level requirements. If the customer is not convinced that the requirements will result in the fulfillment of the security goals, the overall security of the system will be a difficult position for the security engineer to defend.
2) Generate detailed test procedures for security requirements. During verification, record all test results. Use screen prints and save data to files where possible so that results can be reviewed at a later time.
3) Generate a detailed system configuration record for the system used during testing. Compare this record to the planned system configuration for the operational system; explain the differences and their impact on security.
4) Testing should not only inspect and probe low-level details of the system such as attempting a disallowed protocol through the firewall or verifying that a password shadow file is in use, it should also test the application that is expected to run on the secured system. Tests should show that a regular user is able to use the system appropriately and without undue hindrances caused by security measures; for instance, an authorized user is able to log in from a web interface and query the database. Tests should also show data successfully flowing from end to end throughout the entire system and that performance is not hindered by security measures such as a proxy-based firewall.
5) Submit all records of security requirement verification to the security review board for long-term control.

BP.11.01	Identify the solution to be verified and validated.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

BP.11.02	Define the approach and level of rigor for verifying and validating each solution.
BP.11.03	Verify that the solution implements the requirements associated with the previous level of abstraction.
BP.11.04	Validate the solution by showing that it satisfies the needs associated with the previous level of abstraction, ultimately meeting the customer's operational security needs.
BP.11.05	Capture the verification and validation results for the other engineering groups.

The goal of PA11 is to assess the appropriateness of the security solution from two perspectives: verify that the solution that was implemented is exactly what was planned, and validate that the solution that was planned will fully accomplish the security goals of the customer. Failure on either one of these counts results in a system that does not meet the customer's needs, so it is important to investigate these two questions. For example, for a security goal of taking all reasonable steps to avoid malicious software, there might be a requirement to filter incoming email for viruses. The requirement could be verified by inspecting the configuration of the mail server. But this requirement would need to work in conjunction with other requirements in order to be validated as meeting the security goal. Some of those other requirements might include policies eliminating the use of personally-owned software or media, a ban on downloading executables from the Internet, or standards for virus detection software on individual computers.

PA11 deals with the applicability and verification of the requirements themselves, so it should result in requirements that call for both an overall system view of the security efforts and a plan for ensuring that those efforts were successful. Some example requirements might include the following.

1) For each requirement, create a verification plan that describes the method of testing (demonstration, analysis, inspection, or test) as well as a somewhat detailed description of the test to be run. The verification plan might also include a list of the hardware and software packages to be used in the test. Present this plan to the security review board for approval.
2) Document the requirements that are associated with each security goal. Explain how the requirements work together to achieve the goal while still allowing the necessary functionality of the system to continue. Present the documentation to the customer and obtain approval for the chosen implementation.
3) Report results from the verification of performance requirements to the systems engineering organization for further analysis.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Benefits of the SSE-CMM Approach

Using the SSE-CMM as a vehicle to generate security requirements meets the goals discussed earlier – it is a logical approach, and it provides a foundation for future changes.

The SSE-CMM provides a flexible approach that can be molded to fit the security needs of any project. It is not specific about how security issues are to be identified and resolved, but it lays down the principles to follow when doing so. It covers the entire life cycle of the project, from initial architecture decisions to monitoring of the operational system.

The SSE-CMM provides a finite set of areas to consider. Sixty-one base practices is a lot of ground to cover, but the security engineer can have some confidence that once those items are covered, the entire breadth of the security spectrum has been addressed.

The SSE-CMM, as an ISO-accepted standard developed by an industry/government consortium, lends an air of credibility to the security engineer's claims of a comprehensive security implementation. The security plans and other documentation created as a result of the requirements generated from the SSE-CMM will provide the customer with confidence that security will be maintained.

The SSE-CMM provides a ready answer for future increases in funding. There will almost certainly be some base practices that were not implemented initially. And for those base practices that are in place, increased funding can be used to raise the maturity level of those practices by further defining, quantifying, and improving the processes.

The SSE-CMM provides a repeatable approach to the generation of security requirements, which can be valuable when a new threat arises. By assessing the threat, reviewing the architecture, following operational procedures to deal with the threat, and explaining the solution to the customer, the security engineer can reliably handle new requirements just as well as the original ones were handled.

The use of the SSE-CMM provides a clear roadmap to generating security requirements. This roadmap can mean the difference in a never-ending swirl of requirements changes and issues or steady progress towards a robust security requirements baseline.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

Aton, Amy. "Securing Unix Step-by-Step." URL: http://www.giac.org/practical/Amy_Aton_gcux.zip (8 March 2003).

Carnegie Mellon Software Engineering Institute. "Capability Maturity Model for Software (SW-CMM)." 23 January 2003. URL: http://www.sei.cmu.edu/cmm (8 March 2003).

Carnegie Mellon Software Engineering Institute. "Concept of Operations for the CMMI." 11 November 2002. URL: http://www.sei.cmu.edu/cmmi/background/conops.html (8 March 2003).

Cole, Eric; Newfield, Mathew; Millican, John M. SANS GIAC Certification: Security Essentials Toolkit (GSEC). Indianapolis: Que Publishing, 2002. 136 – 139.

Fulton, Bradley. "The Weakest Link: The Human Factor." 29 August 2001. URL: http://www.sans.org/rr/encryption/human.php (8 March 2003).

International Systems Security Engineering Association. "Press Release – Version 2 of the SSE-CMM Accepted by the ISO." 8 August 2002. URL: http://www.issea.org/news/press/press.htm (8 March 2003).

International Systems Security Engineering Association. "The Systems Security Engineering Capability Maturity Model (SSE-CMM)." 25 April 2002. URL: http://www.issea.org/sse_cmm/sse_cmm.html (8 March 2003).

Lowans, Paul. "Implementing a Network Security Metrics Program." URL: http://www.giac.org/practical/Paul_Lowans_GSEC.doc (8 March 2003).

The SANS Institute. "Clarke Says Cyberterrorism is a Real Threat." SANS NewsBites. Volume 5:1 (8 January 2003): Editor's Note. URL: http://www.sans.org/newsletters/newsbites/vol5_1.php (8 March 2003).

SSE-CMM Project. "Systems Security Engineering Capability Maturity Model, Model Description Document Version 2.0." 1 April 1999. URL: http://www.sse-cmm.org/model/ssecmmv2final.pdf (8 March 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix A: Capability Dimension Overview (SSE-CMM, p. 312-314)

Capability Level 1 – Performed Informally

At this level, all base practices are performed somewhere in the project's or organization's implemented process. However, consistent planning and tracking of that performance is missing. Good performance, therefore, depends on individual knowledge and effort. Work products are generally adequate, but quality and efficiency of production depend on how well individuals within the organization perceive that tasks should be performed. Based on experience, there is general assurance that an action will be performed adequately when required. However, the capability to perform an activity is not generally repeatable or transferable.

Common Feature 1.1 – Base Practices Are Performed

GP 1.1.1 – Perform the Process

Capability Level 2 – Planned and Tracked

At the Planned and Tracked level, planning and tracking are introduced. There is general recognition that the organization's performance is dependent on how efficiently the security engineering base practices are implemented within a project's or organization's process. Therefore, work products related to base practice implementation are periodically reviewed and placed under version control. Corrective action is taken when indicated by variances in work products. The primary distinction between the Performed Informally and the Planned and Tracked levels is that at the latter level, the execution of base practices in the project's implemented process is planned and managed. Therefore, it is repeatable within the implementing project, though not necessarily transferable across the organization.

Common Feature 2.1 – Planning Performance

GP 2.1.1 – Allocate Resources
GP 2.1.2 – Assign Responsibilities
GP 2.1.3 – Document the Process
GP 2.1.4 – Provide Tools
GP 2.1.5 – Ensure Training
GP 2.1.6 – Plan the Process

Common Feature 2.2 – Disciplined Performance

GP 2.2.1 – Use Plans, Standards, and Procedures
GP 2.2.2 – Do Configuration Management

Common Feature 2.3 – Verifying Performance

GP 2.3.1 – Verify Process Compliance
GP 2.3.2 – Audit Work Products

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Common Feature 2.4 – Tracking Performance

GP 2.4.1 – Track with Measurement
GP 2.4.2 – Take Corrective Action

Capability Level 3 – Well Defined

At this level, base practices are performed throughout the organization via the use of approved, tailored versions of standard, documented processes. Data from using the process are gathered and used to determine if the process should be modified or improved. This information is used in planning and managing the day-to-day execution of multiple projects within the organization and is used for short- and long-term process improvement. The main difference between the Planned and Tracked and Well Defined levels is the use of organization-wide, accepted standard processes, that implement the characteristics exhibited by the base practices. The capability to perform an activity is, therefore, directly transferable to new projects within the organization.

Common Feature 3.1 – Defining a Standard Process

GP 3.1.1 – Standardize the Process
GP 3.1.2 – Tailor the Standard Process

Common Feature 3.2 – Perform the Defined Process

GP 3.2.1 – Use a Well-Defined Process
GP 3.2.2 – Perform Defect Reviews
GP 3.2.3 – Use Well-Defined Data

Common Feature 3.3 – Coordinate Practices

GP 3.3.1 – Perform Intra-Group Coordination
GP 3.3.2 – Perform Inter-Group Coordination
GP 3.3.3 – Perform External Coordination

Capability Level 4 – Quantitatively Controlled

At the Quantitatively Controlled level, measurable process goals are established for each defined process and associated work products, and detailed measures of performance are collected and analyzed. These data enable quantitative understanding of the process and an improved ability to predict performance. Performance, then, is objectively managed and defects are selectively identified and corrected.

Common Feature 4.1 – Establishing Measurable Quality Goals

GP 4.1.1 – Establish Quality Goals

Common Feature 4.2 – Objectively Managing Performance

GP 4.2.1 – Determine Process Capability
GP 4.2.2 – Use Process Capability

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Capability Level 5 – Continuously Improving

This is the highest achievement level from the viewpoint of process capability. The organization has established quantitative, as well as qualitative, goals for process effectiveness and efficiency, based on long-range business strategies and goals. Continuous process improvement toward achievement of these goals using timely, quantitative performance feedback has been established. Pilot testing of innovative ideas and planned insertion of new technology achieves further enhancements.

Common Feature 5.1 – Improving Organizational Capability

GP 5.1.1 – Establish Process Effectiveness Goals
GP 5.1.2 – Continuously Improve the Standard Process

Common Feature 5.2 – Improving Process Effectiveness

GP 5.2.1 – Perform Causal Analysis
GP 5.2.2 – Eliminate Defect Causes
GP 5.2.3 – Continuously Improve the Defined Process

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix B: Security Engineering Process Area Overview (SSE-CMM, p. 315-318)

The security engineering category groups together those process areas that are primarily concerned with performing security engineering. They are organized alphabetically within the category to discourage the reader from inferring a sequence for the process areas.

PA01: Administer Security Controls

Goal 1	Security controls are properly configured and used.
BP.01.01	Establish responsibilities and accountability for security controls and communicate them to everyone in the organization.
BP.01.02	Manage the configuration of system security controls.
BP.01.03	Manage security awareness, training, and education programs for all users and administrators.
BP.01.04	Manage periodic maintenance and administration of security services and control mechanisms.

PA02: Assess Impact

Goal 1	The security impacts of risks to the system are identified and characterized.
BP.02.01	Identify, analyze, and prioritize operational, business, or mission capabilities leveraged by the system.
BP.02.02	Identify and characterize the system assets that support the key operational capabilities or the security objectives of the system.
BP.02.03	Select the impact metric to be used for this assessment.
BP.02.04	Identify the relationship between the selected metrics for this assessment and metric conversion factors if required.
BP.02.05	Identify and characterize impacts.
BP.02.06	Monitor ongoing changes in the impacts.

PA03: Assess Security Risk

Goal 1	An understanding of the security risk associated with operating the system within a defined environment is achieved.
Goal 2	Risks are prioritized according to a defined methodology.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

BP.03.01	Select the methods, techniques, and criteria by which security risks, for the system in a defined environment are analyzed, assessed, and compared.
BP.03.02	Identify threat/vulnerability/impact triples (exposures).
BP.03.03	Assess the risk associated with the occurrence of an exposure.
BP.03.04	Assess the total uncertainty associated with the risk for the exposure.
BP.03.05	Order risks by priority.
BP.03.06	Monitor ongoing changes in the risk spectrum and changes to their characteristics.

PA04: Assess Threat

Goal 1	Threats to the security of the system are identified and characterized.
BP.04.01	Identify applicable threats arising from a natural source.
BP.04.02	Identify applicable threats arising from man-made sources, either accidental or deliberate.
BP.04.03	Identify appropriate units of measure, and applicable ranges, in a specified environment.
BP.04.04	Assess capability and motivation of threat agent for threats arising from man-made sources.
BP.04.05	Assess the likelihood of an occurrence of a threat event.
BP.04.06	Monitor ongoing changes in the threat spectrum and changes to their characteristics.

PA05: Assess Vulnerability

Goal 1	An understanding of system security vulnerabilities within a defined environment is achieved.
BP.05.01	Select the methods, techniques, and criteria by which security system vulnerabilities in a defined environment are identified and characterized.
BP.05.02	Identify system security vulnerabilities.
BP.05.03	Gather data related to the properties of the vulnerabilities.
BP.05.04	Assess the system vulnerability and aggregate vulnerabilities that result from specific vulnerabilities and combinations of specific vulnerabilities.
BP.05.05	Monitor ongoing changes in the applicable vulnerabilities and changes to their characteristics.

PA06: Build Assurance Argument

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Goal 1	The work products and processes clearly provide the evidence that the customer's security needs have been met.
BP.06.01	Identify the security assurance objectives.
BP.06.02	Define a security assurance strategy to address all assurance objectives.
BP.06.03	Identify and control security assurance evidence.
BP.06.04	Perform analysis of security assurance evidence.
BP.06.05	Provide a security assurance argument that demonstrates the customer's security needs are met.

PA07: Coordinate Security

Goal 1	All members of the project team are aware of and involved with security engineering activities to the extent necessary to perform their functions.
Goal 2	Decisions and recommendations related to security are communicated and coordinated.
BP.07.01	Define security engineering coordination objectives and relationships.
BP.07.02	Identify coordination mechanisms for security engineering.
BP.07.03	Facilitate security engineering coordination.
BP.07.04	Use the identified mechanisms to coordinate decisions and recommendations related to security.

PA08: Monitor Security Posture

Goal 1	Both internal and external security related events are detected and tracked.
Goal 2	Incidents are responded to in accordance with policy.
Goal 3	Changes to the operational security posture are identified and handled in accordance with the security objectives.
BP.08.01	Analyze event records to determine the cause of an event, how it proceeded, and likely future events.
BP.08.02	Monitor changes in threats, vulnerabilities, impacts, risks, and the environment.
BP.08.03	Identify security relevant incidents.
BP.08.04	Monitor the performance and functional effectiveness of security safeguards.
BP.08.05	Review the security posture of the system to identify necessary changes.
BP.08.06	Manage the response to security relevant incidents.
BP.08.07	Ensure that the artifacts related to security monitoring are suitably protected.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

PA09: Provide Security Input

Goal 1	All system issues are reviewed for security implications and are resolved in accordance with security goals.
Goal 2	All members of the project team have an understanding of security so they can perform their functions.
Goal 3	The solution reflects the security input provided.
BP.09.01	Work with designers, developers, and users to ensure that appropriate parties have a common understanding of security input needs.
BP.09.02	Determine the security constraints and considerations needed to make informed engineering choices.
BP.09.03	Identify alternative solutions to security related engineering problems.
BP.09.04	Analyze and prioritize engineering alternatives using security constraints and considerations.
BP.09.05	Provide security related guidance to the other engineering groups.
BP.09.06	Provide security related guidance to operational system users and administrators.

PA10: Specify Security Needs

Goal 1	A common understanding of security needs is reached between all parties, including the customer.
BP.10.01	Gain an understanding of the customer's security needs.
BP.10.02	Identify the laws, policies, standards, external influences and constraints that govern the system.
BP.10.03	Identify the purpose of the system in order to determine the security context.
BP.10.04	Capture a high-level security oriented view of the system operation.
BP.10.05	Capture high-level goals that define the security of the system.
BP.10.06	Define a consistent set of statements which define the protection to be implemented in the system.
BP.10.07	Obtain agreement that the specified security meets the customer's needs.

PA11: Verify and Validate Security

Goal 1	Solutions meet security requirements.
Goal 2	Solutions meet the customer's operational security needs.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

BP.11.01	Identify the solution to be verified and validated.
BP.11.02	Define the approach and level of rigor for verifying and validating each solution.
BP.11.03	Verify that the solution implements the requirements associated with the previous level of abstraction.
BP.11.04	Validate the solution by showing that it satisfies the needs associated with the previous level of abstraction, ultimately meeting the customer's operational security needs.
BP.11.05	Capture the verification and validation results for the other engineering groups.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 26th, 2013

Upcoming SANS Training

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

SANS Malaysia @ MCMC 2013	**Cyberjaya, MY**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
SANS Pen Test Berlin 2013	**Berlin, DE**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
Industial Control Systems Security Training - Houston	**Houston, TXUS**	**Jun 10, 2013 - Jun 15, 2013**	**Live Event**
Security Impact of IPv6 Summit 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 16, 2013**	**Live Event**
SANSFIRE 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 22, 2013**	**Live Event**
SANS Canberra 2013	**Canberra, AU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
Mobile Device Security Summit 2013	**OnlineCAUS**	**May 30, 2013 - Jun 06, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**