SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

Security Engineering Training

A Framework for Corporate Training Programs on the Principles of Secure Software Development

April 20, 2009

Editor Stacy Simpson, SAFECode

Contributors

Eric Baize, EMC Corporation
Reeny Sondhi, EMC Corporation
Hardik Parekh, EMC Corporation
Dan Reddy, EMC Corporation
Brad Minnis, Juniper Networks, Inc.
Bernie Rosen, Juniper Networks, Inc.
Michael Howard, Microsoft, Corp.
Steve Lipner, Microsoft Corp.
Glenn Pittaway, Microsoft Corp.
Antti Vähä-Sipilä, Nokia
Cassio Goldschmidt, Symantec Corp.
Wesley Higaki, Symantec Corp.
Paul Kurtz, SAFECode

Table of Contents

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

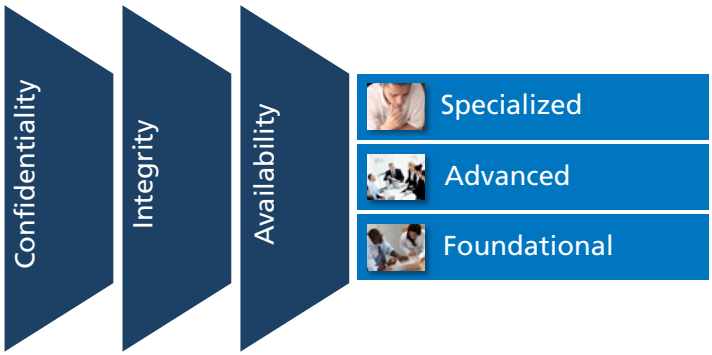
Introduction

Software assurance plays a vital role in protecting the information infrastructure, giving technology vendors both a responsibility and business incentive to improve the security of the software they produce. Recognizing this, many information and communications technology leaders are developing internal software assurance programs to reduce vulnerabilities, improve resistance to attack and protect the integrity of software. Fundamental to the success of these programs is the ability to ensure that the people designing, developing and testing products understand the fundamentals of secure engineering.

> Software assurance encompasses methods and processes that ensure software functions as intended while mitigating the risks of vulnerabilities and malicious code that could bring harm to the end user.

In an analysis of the software assurance programs of SAFECode members, it quickly becomes evident that each successful effort has been supported by internally-developed security engineering training directed at all those responsible for the development of the software they produce, including product managers, project managers, architects/designers, developers and testers. The need for in-house training is partly due to the fact that secure development principles are not yet a significant part of the software engineering curriculum at the college and university level. While a small number of universities are working to add secure design principles to the programming curriculum, these initiatives are still in their infancy. Moreover, internally-developed training is the only way to build the specialized skills and knowledge necessary for supporting an organization's unique development environment, processes and security policies. As such, SAFECode recommends that security engineering training be considered as a part of any software assurance program since managers cannot assume that their product teams already have the skills needed to effectively implement secure development principles.

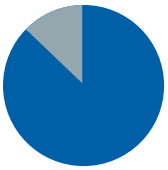
This paper outlines the fundamentals of a security engineering training program based on an analysis of the shared experiences of SAFECode members. It is not meant to provide a curriculum, but rather a framework that can be put into place to facilitate successful training initiatives across diverse corporate cultures, development environments and product requirements. While SAFECode recognizes that building an in-house training program can be a challenge in smaller organizations, its hope is that organizations of all sizes will find value in tailoring many of the principles of the framework to meet their individual requirements.



Three levels of security engineering training

A Framework for Internal Security Engineering Training

A qualitative 2008 survey by the Cyber Security Knowledge Transfer Network concluded that fewer than 20 percent of UK computing undergraduates get a meaningful education in secure development and design.



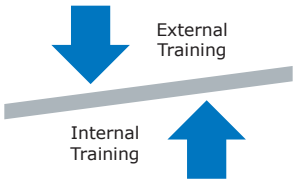
The decision to create an in-house training program versus outsourcing training or building teams that already possess desired skill sets is not taken lightly. Building an effective internal training program requires a significant investment of resources. However, there are numerous reasons why an internally developed program is the favored – and in many ways the required – approach of SAFECode members.

The lack of formal education on secure software design, development and testing principles at the university level and the infancy of many corporate software assurance programs have resulted in a shortage

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

of software engineers who already possess the secure development skills desired by software vendors. This makes it extremely difficult to build teams already fully educated on secure development practices. For this reason, supporting some level of training to supplement the security engineering skills of product teams is a requirement for nearly every organization implementing software assurance programs.

Once it becomes clear that some level of corporate-sponsored training is required, the first instinct is often to look to outsource training initiatives or obtain some industry standard curriculum to use internally. However, even when outside training programs are leveraged or other curriculums adapted, it must be recognized that they will not directly relate to an organization's unique development environment, processes and security policies. As such, some additional instruction tailored to the corporate environment is still necessary.



It should be noted in this context that there are a number of secure software development training and certification programs available that can help advance the security skill sets of individuals and bring knowledge back into the workplace. While these programs are not a replacement for corporate in-house training programs, they do provide software engineering professionals an opportunity to advance and validate their skills and should be considered on an individual basis, especially for those wishing to advance their careers.

List of Security Engineering Training and Certification Programs

There are many outside training programs that fill an important need for specialized technical training for security practitioners and certain segments of development teams, a number of which provide professional certification credentials. While SAFECode does not endorse any one program or approach, examples of some of some of these programs are provided below. Included are courses directed at information security professionals and software development engineers, so the focus on secure development practices varies greatly between programs, however, elements of both types of training programs may be applicable depending on an individual's background and specialized needs.

Programs Primarily Focused on Secure Software Development

Certified Secure Software Lifecycle Professional (CSSLP)

www.isc2.org

Developed by the International Information Systems Security Certification Consortium, Inc. [(ISC)2], the Certified Secure Software Lifecycle Professional (CSSLP) is a certification designed to validate secure software development knowledge and expertise.

EC-Council's Certified Secure Programmer and Certified Secure Application Developer

www.eccouncil.org/ECSP.htm

EC-Council's Certified Secure Programmer and Certified Secure Application Developer programs aim to provide the essential and fundamental skills to programmers and application developers in secure programming.

GIAC Secure Software Programmer (GSSP) Certification

www.giac.org/certifications/software/

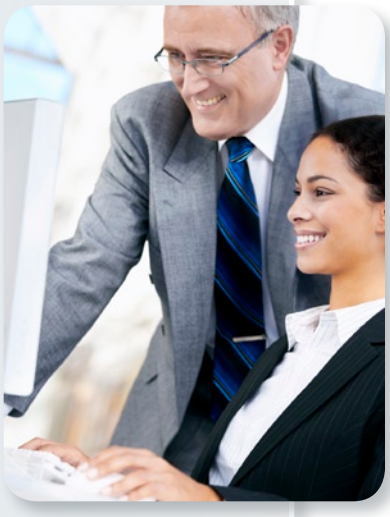
The GIAC Secure Software Programmer (GSSP) Certification Exam was developed in a joint effort involving the SANS Institute, CERT/CC, several US government agencies, and leading companies in the US, Japan, India, and Germany. It allows candidates to demonstrate mastery of the security knowledge and skills needed to deal with common programming errors that lead to most security problems.

ISSECO Certified Professional for Secure Software Engineering

www.isseco.org

ISSECO (International Secure Software Engineering Council) offers the Certified Professional for Secure Software Engineering program that focuses on providing and validating the skills necessary to produce secure software.

Programs Primarily Focused on Information Security Management and Operations



Certified Information Security Manager (CISM)

www.isaca.org

Developed by ISACA, CISM is a certification program is developed specifically for experienced information security managers and those who have information security management responsibilities. The CISM certification aims to enhance and validate the skills of the individual who manages, designs, oversees and/or assesses an enterprise's information security.

Certified Information Systems Security Professional (CISSP)

www.isc2.org

Developed by the International Information Systems Security Certification Consortium, Inc. [(ISC)2], CISSP is a credential accredited by ANSI to ISO Standard 17024:2003 in the field of information security. It is aimed at providing information security professionals with an objective measure of competence.

CompTIA Security+™ Certification

certification.comptia.org/security/

CompTIA Security+ validates knowledge of systems security, network infrastructure, access control, assessments and audits, cryptography and organizational security.

EC-Council's Ethical Hacker

www.eccouncil.org/ceh.htm

The CEH Program certifies individuals in the specific network security discipline of Ethical Hacking from a vendor-neutral perspective. Certified Ethical Hacker is a skilled professional who understands and knows how to look for the weaknesses and vulnerabilities in target systems and uses the same knowledge and tools as a malicious hacker.

Member of the Institute of Information Security Professionals (M.Inst.ISP)

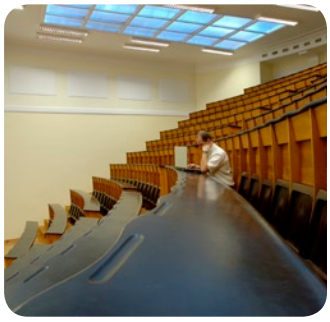
www.instisp.org

The Institute of Information Security Professionals is relatively new initiative, originating from the UK, that through membership accredits security professional competency by peer review of an individual's practical application of knowledge in a similar manner to other professions such as engineers.

The final, and perhaps most important, reason that SAFECode's members favor an in-house approach to security engineering training is the ability to tie their training initiatives in a concrete way to corporate goals, processes and risk management approaches, as well as employee performance expectations. In soliciting feedback from employees undergoing training, it is evident that the training is most embraced when the employees can directly apply what they learn to their daily work. In this way, training becomes more than an abstract corporate requirement, but rather a tool they can use to continue their professional development and further their careers. While this is very much a positive and a common trait in corporate training programs regardless of subject matter, it can create challenges for the development of an industry standard curriculum due not only to the diversity in corporate cultures, development environments and product requirements, but also the nascent nature of formal software assurance programs.

For these reasons, the framework that SAFECode is offering for internal software engineering training programs provides an approach that can be tailored and adapted across diverse corporate environments. Companies can use the framework to focus on the knowledge and skills that are most important to their needs of their programs, and thus meet their corporate objectives.

It should be noted that SAFECode believes that industry must advocate for formalized security engineering education at the college and university level and hopes that as software assurance programs advance, a more standardized curriculum can be developed for both full-time programs and ongoing continuing education. However, corporations cannot wait for these developments to occur before integrating secure development principles into their development lifecycles and it is our experience that this knowledge gap can be addressed through internally-developed training initiatives.



Define Training Targets and Learning Goals

Target a Broad Audience

The most important theme that arose from the analysis of the training programs of SAFECode members was the importance of setting a solid base of foundational knowledge across the entire product team – Product Managers, Product Architects/Designers, Technical Writers, Program/Project Managers, Development Engineers, Service and Quality Assurance (QA) Engineers. It is imperative that awareness training be given to everyone who touches product development in order to build a more security-aware culture. Once this mindset is achieved, it becomes easier to change the specific behaviors of developers and QA professionals.

Develop Security Advocates Throughout the Organization to Spread the Word

Companies have found that a centralized program cannot be successful alone without security knowledgeable and trained professionals embedded within the organization. Through a variety of means member companies have found ways of building an internal network of security advocates who can extend what they learn within their local teams.

> Conceptual understanding of security issues, including buffer overflows, data validation, SQL injection, cross site scripting, format string vulnerabilities and use of unsafe functions or behaviors, etc., are also important skills for developers and QA engineers.

To meet these goals, content should create an understanding of basic security such as

- The current threat environment and the corresponding importance of secure development practices
- Secure design principles
- Secure coding principles
- The most common errors that lead to security vulnerabilities
- Threat modeling
- How to find/test security-related issues in code
- How to fix security issues, etc.
- Security in the software development lifecycle (of your company)

Develop or Obtain Training Content within the Framework

Again, while not a detailed curriculum, SAFECode does have some recommendations regarding what course material a comprehensive security engineering training program should cover.

Courses can be broken into three main levels: Foundational, Advanced and Specialized.

Foundational

Foundational courses designed to raise the level of security awareness should be provided to all who touch the product development process – product/project/program managers, team leads, developers, QA and sustaining engineers. This level of training is essential in building a development culture that prioritizes security and further enables the success of software assurance programs. Managers must understand the importance of security and basic security issues or developers and QA engineers will not be supported in their efforts to advance their secure coding or testing skills and leverage secure development practices in the course of their daily work. In effect, all team members should have a basic understanding of the current threat environment, i.e. the increasingly organized and targeted nature

of cybercrime, examples of previous vulnerability exploits, etc., and the important role of secure development practices in attack prevention. As such, foundational courses should include introductory courses on product security topics, business rationale, mindset of security as an essential software characteristic and the basics of secure coding and testing.

Advanced

Advanced training courses are where the majority of secure coding and testing practices are taught. It is important that this training is role-based, meaning that developers and QA engineers should receive training directly applicable to their job function and the company's security development lifecycle. Courses should cover topics such as language-specific practical techniques to prevent and fix software vulnerabilities, defensive coding techniques, detailed overview of a company's security development lifecycle, threat modeling, how to find security flaws in software and how to fix them.

Specialized

Additional training should be available to developers and QA engineers seeking to improve their security knowledge in specialized areas. These trainings are typically role-based and directly tied to job functions. Examples of topics covered include training on specific static code analysis tools and cryptography.

The charts to the right provide examples of courses taught by SAFECode members in each of these categories.

While some of the skills/knowledge required are static and able to be adequately covered as a one-time exercise, other aspects do evolve and require follow-up. It is important to develop an understanding of the ever-changing, dynamic threat landscape that enables developers to evaluate the software they develop constantly and their knowledge of security issues based on the changes in the attack environment. Many SAFECode members supplement their internal training programs with informal approaches to continuing education through podcasts, newsletters, in-house conferences, webinars, etc. that keep product teams updated on security developments.

Examples of Internally-developed Security Engineering Classes Offered by SAFECode Members

Foundational: Intended for all employees involved in product development and management.

Role	Title	Format
Everyone	Business level understanding of why security is important and how to address it	Computer-based Training
Product/Program Managers, Dev Leads, Dev Engineers, QA Engineers	Security Engineering Principles	Computer-based Training
Product/Program Managers, Dev Leads, Dev Engineers, QA Engineers	Security in the Software Development Lifecycle	Computer-based Training

Advanced: Intended for all employees involved in product development and testing.

Role	Title	Format
Dev Engineers	Language- and Environment-specific Secure Coding	Combination of Instructor-led and Computer-based Training
QA Engineers	Security Testing	Combination of Instructor-led and Computer-based Training
Dev Engineers, Architects	Introduction to Threat Modeling	Computer-based Training
Dev Engineers, Architects	Secure Design Principles	Computer-based Training

Specialized: Intended for engineers seeking to improve their security knowledge in specialized areas.

Role	Title	Format
Dev Engineers, QA Engineers	Security Tools Training (includes static code analysis tools, security testing tools, etc.)	Computer-based Training
Dev Engineers	Specific technology implementations and security concerns created or influenced by those technologies (examples include web application security, Operating System specific security, hardware security, etc.)	Combination of Instructor-led and Computer-based Training
Dev Engineers, QA Engineers	Common vulnerabilities, how they can be exploited, how to avoid them during development and how to test for them	Combination of Instructor-led and Computer-based Training
Dev Engineers, Architects	Introduction to Cryptography	Computer-based Training

Determine How Training Program will be Implemented

No training program can be effective if is not embraced by its participants. While security engineering basics should be considered part of the required skill set for product teams, determining whether or not to mandate security engineering training should be dependent on an organization's culture, approach to other professional development initiatives and the level of knowledge its staff currently holds. However, regardless of whether they require security training participation, all SAFECode members have worked diligently to build a development culture that prioritizes security, providing teams and individuals with a sense of accountability for their contribution to corporate software assurance efforts. In this spirit, each member rewards engineers for their ability to produce more secure software, be it through incentives or professional advancement.

While SAFECode member companies have not faced a significant amount of resistance from training participants, the employee concern most commonly voiced is that they do not have time available to dedicate to training. This is not surprising given the time-to-market pressure that most software development teams face in the fast-paced information technology industry. There are a number of ways SAFECode members have successfully addressed this concern. First, as referenced previously, employee feedback reveals that the customized nature of the training, which provides a direct link between training content and performance expectations, minimizes the chance that learners may perceive time spent in training as wasted. Another strong motivator is the ability to demonstrate that being able to apply secure engineering techniques during coding is less time consuming than going back to fix mistakes later in the process.

In addition, the way in which training is implemented has a significant impact on whether or not it is seen as too time consuming. For example, being mindful of product release schedules when planning courses is recommended so that learners are not asked to take time out for training while they are

facing major corporate deadlines. In addition, taking advantage of computer-based training provides the learner with more control over how they manage their time. For instructor-led courses, careful decision-making over whether to break course time into small chunks or hold full-day sessions is also recommended.

Determining whether course material should be delivered via computer-based training (CBT) or instructor-led training (ILT) is dependent upon a company's unique attributes, size, culture, distribution, etc. For instance, in a smaller company it may actually take more resources to put together an effective CBT program than to bring staff together for instructor-led sessions while a large, global organization will likely find CBT far more efficient and cost effective than organizing seminars across numerous offices.

Each delivery model has advantages and disadvantages. CBT allows training participants to complete courses on their own schedules and at their own pace. In addition, CBT is usually more cost-effective than ILT, especially when there is a need to train a highly distributed workforce. However, CBT may not offer the direct interaction or hands-on lab scenario of ILT.

A number of advantages are provided by ILT including direct interaction with learners, which allows for the immediate answering of questions and a reduction in the likelihood for misunderstanding of course content. In addition, when instructors are senior engineers within the company, a rapport can be built that carries on after the class has ended, allowing for ongoing mentoring. However, ILT can be more expensive, especially if an organization has to turn to outside instructors either because they do not have the expertise in-house or the individuals with the expertise do not have sufficient time to commit to instruction. Implementing an ILT approach with in-house experts also runs the risk of "burning out" instructors if there are a large number of engineers to train on an ongoing basis. It is also challenging to schedule courses in such a way as to guarantee full attendance, especially in larger, more distributed organizations.

Computer-based	Instructor-led
Flexible schedule	Get direct answers
Cost effective	May accommodate labs
Suited for global orgs	In-house mentoring
Can intermix COTS content	Build peer resources

Most SAFECode member companies take a hybrid approach, offering many of the foundational courses via CBT and then supplementing those with ILT for more specialized training for segments of the product teams.

Monitor Results

Measuring the success of a training program is an inexact art form, and as such SAFECode members approach it from different perspectives.

There are direct measurements of training, which include metrics like the number of team members trained and qualitative tools like post-lesson surveys to see how learners are responding to the program. Some members also include detailed training plans as part of employee performance appraisals and track progress against team and individual training goals.

Others rely more heavily on indirect measurements such as the quality of the output of the code, i.e. counting the number of standards violations found during testing or post-release. By evaluating the quality of the code, it is often possible to infer whether or not a training program is effectively getting the right messages across, and more importantly, resulting in the desired behavioral changes amongst participants. One member compared the number of trained engineers on product teams and found that products with a higher number of trained engineers consistently perform better in penetration tests than those with whose teams had fewer trained engineers coding on the same platform and with similar tools.

Regardless of the specific methods used, some form of training programs assessment is recommended. Being able to point to positive results will not only help to maintain management's commitment to the program, but will also help further motivate engineers to participate. Obtaining feedback on the training program from its participants through post-class surveys and other mechanisms will also allow for continues improvement.

Future Directions

A workforce skilled in software assurance practices is essential to the efforts to improve the overall state of software security and its development must be a priority for all key stakeholders. While in-house security engineering training is a critical element of corporate software assurance programs and a step in the right direction, it is important to underscore that it is not a replacement for formal education on secure development principles and practices at the university level.

It is imperative that the software industry continues to support colleges and universities in their efforts to include basic software assurance practices in their software engineering curriculum. This will help to ensure that future generations of software engineers have a basic understanding secure development principles and practices and, equally as important, view them as a core component of the software engineering discipline. Further, a more skilled workforce will allow corporations to focus on fine-tuning the skills of their development teams to better meet their specific organizational and customer requirements to both the industry's and its customers' benefit.

Conclusion

There is wide agreement that software assurance plays a vital role in protecting the information infrastructure. And yet, few software engineering professionals receive any formal training on secure design, development and testing principles. The lack of security engineering awareness and education among the software engineering workforce is a significant obstacle to information and communications technology corporations working to implement effective software assurance programs.

All of SAFECode's member companies have had to face this issue in their own software assurance efforts and all of them had to implement an in-house security engineering training program to overcome it. It is widely recognized that universities must step up their efforts to provide more education on secure design, development and testing methods to their software engineering students. However, as much as industry must continue to advocate and support universities in their efforts to make this a reality, the simple fact is that it cannot afford to wait for it to happen before taking action. Based on the shared experiences of SAFECode member companies, the most effective means of building a workforce with the skills necessary to effectively implement software assurance initiatives is through an investment in internally-driven security engineering training programs.

While the experiences of SAFECode's member companies demonstrate that internal training programs are most effective when customized to unique corporate needs, the programs do share some foundational elements that can greatly contribute to overall success. It is SAFECode's hope that by collecting, analyzing and sharing these elements, it can provide others in the industry with a useful framework for building their own internal training programs.

About SAFECode

The Software Assurance Forum for Excellence in Code (SAFECode) is a non-profit organization exclusively dedicated to increasing trust in information and communications technology products and services through the advancement of effective software assurance methods. SAFECode is a global, industry-led effort to identify and promote best practices for developing and delivering more secure and reliable software, hardware and services. Its members include EMC Corporation, Juniper Networks, Inc., Microsoft Corp., Nokia, SAP AG and Symantec Corp. For more information, please visit www.safecode.org.

SAFECode
2101 Wilson Boulevard
Suite 1000
Arlington, VA 22201

(p) 703.812.9199
(f) 703.812.9350
(email) stacy@safecode.org
www.safecode.org

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity