

Cybersecurity Standards and Frameworks

CS-3113: PRINCIPLES OF CYBER SECURITY

BENJAMIN R. ANDERSON

Background

Not all facets of security involve hardware and software

Other facets that need to be considered:

- Policies and Procedures
- Roles and Responsibilities
- Resource allocation
- Business functions
- Legal, regulatory, and other requirements (HIPAA, PII, trade secrets, etc.)
- Many others

A viable security program needs to consider them all

- If not, something will be overlooked and exploited!

Background

There have been many different standards and frameworks that have been developed to address cybersecurity within an organization

The creators of these standards and frameworks often have different “audiences”

- Some, such as NIST, are specific to a country – for NIST, that is the U.S.
- Others, such as ISO standards, are international
 - **ISO**: International Organization for Standardization
- Others might be related to a specific industry or task
 - PCI DSS – The Payment Card Industry, Data Security Standard applies to organizations that handle credit card transactions
 - BSIMM – The Building Security In Maturity Model is for developing secure software

There are quite a few of these standards, and some overlap

- For example, a government agency that processes credit cards would have to follow NIST and PCI DSS

Purpose of Frameworks and Standards

The field of cybersecurity is very, very broad

For an organization that is just starting a cybersecurity program, they may not know where to even begin

Frameworks can help address:

- What areas to include in a security program
- What aspects of security need to be considered
- What the critical security functions are
- What order security functions need to be implemented
- Understanding what is actually necessary

Organisation for Economic Co-operation and Development (OECD)

OECD is an “intergovernmental economic organization” with 38 member countries

One of the roles of the OECD is to identify best practices

One of their publications is:

- Digital Security Risk Management for Economic and Social Prosperity
- Published in 2015
- <https://www.oecd.org/digital/ieconomy/digital-security-risk-management.pdf>

The purpose of this document (from the Foreward):

- *This OECD Recommendation on Digital Security Risk Management for Economic and Social Prosperity and its Companion Document provide guidance for a new generation of national strategies on the management of digital security risk aimed to optimize the economic and social benefits expected from digital openness.*

This document provides four general, and four operational principles

OECD General Principles

Principles	Description
Awareness, skills, and empowerment	All stakeholders should understand digital security risk and how to manage it.
Responsibility	All stakeholders should take responsibility for the management of digital security risk.
Human rights and fundamental values	All stakeholders should manage digital security risk in a transparent manner and consistently with human rights and fundamental values.
Co-operation	All stakeholders should co-operate, including across borders.

From: *Digital Security Risk Management for Economic and Social Prosperity*
<https://www.oecd.org/digital/ieconomy/digital-security-risk-management.pdf>

OECD Operational Principles

Principles	Description
Risk assessment and treatment cycle	Leaders and decision makers should ensure that digital security risk is treated on the basis of continuous risk assessment.
Security measures	Leaders and decision makers should ensure that security measures are appropriate to and commensurate with the risk.
Innovation	Leaders and decision makers should ensure that innovation is considered.
Preparedness and continuity	Leaders and decision makers should ensure that a preparedness and continuity plan is adopted.

From: *Digital Security Risk Management for Economic and Social Prosperity*
<https://www.oecd.org/digital/ieconomy/digital-security-risk-management.pdf>

ISO 27001

Unfortunately, unlike NIST, these are not free standards, and cannot readily be referenced without purchasing them

The purpose of ISO 27001 is to provide a universal methodology for:

- ***The implementation, management, and maintenance of information security within a company.***

An organization seeks to get ISO 27001 certified to demonstrate:

- Their ***Information Security Management System*** (ISMS) comply with documented standards
- This is often used to meet contractual obligations – especially for those involving government or critical infrastructures
- Other reasons for getting certified:
 - Demonstrate the maturity of their information security environment
 - Gain a competitive uniqueness against their competition."

This standard adopts the "***Plan-Do-Check-Act***" (PDCA) model, which is applied to all

ISO 27001 – PDCA Model

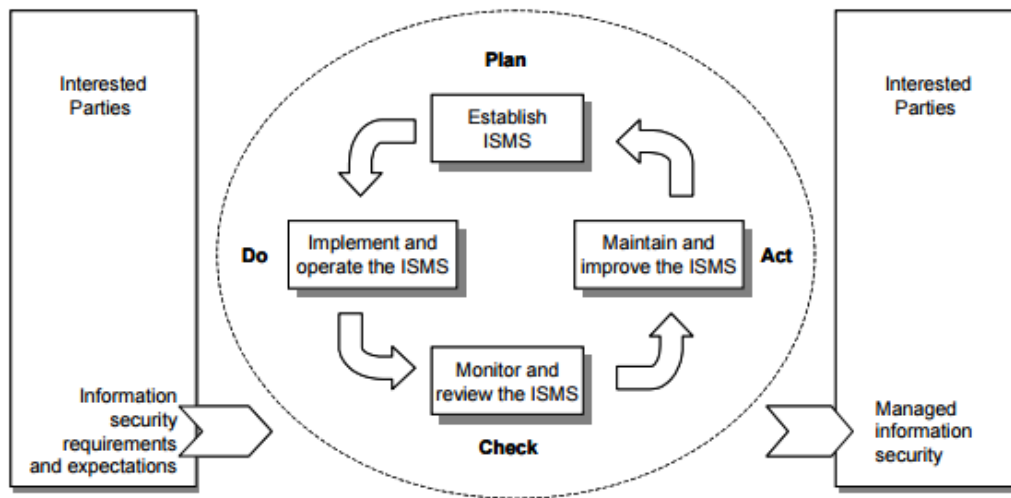


Figure 1 — PDCA model applied to ISMS processes

Plan: Establish the ISMS

- Establish ISMS policy, objectives, processes, and procedures relevant to managing risk and improving information security to deliver results in accordance with an organization's overall policies and objectives.

Do: Implement and operate the ISMS

- Implement and operate the ISMS policy, controls, processes, and procedures.

Check: Monitor and review the ISMS

- Assess and, where applicable, measure process performance against ISMS policy, objectives and practical experience and report the results to management for review.

Act: Maintain and improve the ISMS

- Take corrective and preventive actions, based on the results of the internal ISMS audit and management review or other relevant information, to achieve continual improvement of the ISMS.

Reference: ISO 27001 from 2005

COBIT

From Wikipedia:

- *COBIT (Control Objectives for Information and Related Technologies) is a framework created by ISACA for information technology (IT) management and IT governance.*
- <https://en.wikipedia.org/wiki/COBIT>

The components from COBIT 5 (this has been superseded by COBIT 2019) from Wikipedia:

- **Framework:** Organizes IT governance objectives and good practices by IT domains and processes and links them to business requirements.
- **Process descriptions:** A reference process model and common language for everyone in an organization. The processes map to responsibility areas of plan, build, run, and monitor.
- **Control objectives:** Provides a complete set of high-level requirements to be considered by management for effective control of each IT process.
- **Management guidelines:** Helps assign responsibility, agree on objectives, measure performance, and illustrate interrelationship with other processes.
- **Maturity models:** Assesses maturity and capability per process and helps to address gaps.

COBIT2019

Watch: V1.0 COBIT2019 Overview by Mark Thomas

- <https://www.youtube.com/watch?v=Zq5ZHIPs3TI>

Note that there are now two sets of principles:

- **Governance System Principles:** provide stakeholder value, holistic approach, dynamic governance system, governance distinct from management, tailored to enterprise needs, end-to-end governance system
- **Governance Framework Principles:** Based on conceptual model, open and flexible, aligned to major standards

NIST Cybersecurity Framework: Background

In February 2013, President Obama issued Executive Order (EO) 13636, *Improving Critical Infrastructure Cybersecurity*

- This EO directed NIST to develop a voluntary framework for reducing cyber risks to critical infrastructure

The NIST Cybersecurity Framework (CSF) was the result of this effort

From NIST:

- *NIST initially produced the Framework in 2014 and updated it in April 2018 with CSF 1.1. Based on stakeholder feedback, in order to reflect the ever-evolving cybersecurity landscape and to help organizations more easily and effectively manage cybersecurity risk, NIST is planning a new, more significant update to the Framework: CSF 2.0.*
- <https://www.nist.gov/cyberframework/updating-nist-cybersecurity-framework-journey-csf-20>

Unlike commercial frameworks, the NIST CSF can be found on their website:

- <https://www.nist.gov/cyberframework>

NIST Cybersecurity Framework

The CSF is incredibly large and detailed

- **Watch:** The Cybersecurity Framework, NIST
 - <https://www.youtube.com/watch?v=J9ToNuwmyF0>

Fortunately, NIST provides a number of materials that can assist in learning about and understanding the CSF

- **Review:** The Cybersecurity Framework, Version 1.1, October 2019
 - Make note of the Primary Components: Core, Profiles, and Implementation tiers
 - The components of the Framework Core: Identify, Protect, Detect, Respond, Recover
 - The structure of the Framework: Function, Category, Subcategory, and Informative References
 - The implementation tiers: Partial, Risk Informed, Repeatable, and Adaptive
 - <https://www.nist.gov/document/cybersecurityframeworkv1-1presentationpptx>
- **Read:** NIST Special Publication 1271: Getting Started with the NIST Cybersecurity Framework
 - <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1271.pdf>

NIST Cybersecurity Framework

Read: *Framework for Improving Critical Infrastructure Cybersecurity Version 1.1*, National Institute of Standards and Technology, April 16, 2018

- Sections 1.0 to 4.0 (pages 1-21)
- <https://www.nist.gov/cyberframework/framework>
- Direct link to PDF: <https://doi.org/10.6028/NIST.CSWP.04162018>
- You are responsible for the content in those sections

From the CSF, pay special attention to:

- The components of the Framework: Core, Tiers, and Profiles
- Definitions of Functions, Categories, Subcategories, and Informative References
- The definitions of the 5 Core Functions
- The tier definitions and components
- The definition of a Framework Profile
- How information flows in an organization using the framework
- How the CSF supports supply chain risk management (SCRM)

NIST CSF

In addition, look over the appendices for the CSF

Note how the Functions and Categories fit together

- See the diagram to the right

Also, select a few of the entries in Table 2, and see the level of detail in this framework

- You are not expected to memorize any of these entries, but you should be familiar with how the components of each entry relate

April 16, 2018

Cybersecurity Framework

Version 1.1

Table 1: Function and Category Unique Identifiers

Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

NIST CSF Informative Reference

The reason the CSF only provides references is the sheer size of the referenced material

For example, one reference is NIST SP 800-53 Rev. 4: *Security and Privacy Controls for Information Systems and Organizations*

- SP800-53 is now on Rev. 5, which is 492 pages
- Contains over 1000 controls

If the CSF were to include the references from CIS CSC, COBIT 5, ISA 62443-2-1:2009, ISO/IEC 27001:2013 and NIST SP 800-53 Rev. 4 it would be thousands of pages, and unusable as a practical reference

Software Security

Quote from Dr. Gene Spafford:

- *The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.*
- *Computer Recreations: Of Worms, Viruses and Core War* by A. K. Dewdney in Scientific American, March 1989, pp 110.

While it is possible to program trivial programs that are “provably correct” it is not possible for the general case

- There is just not enough time or resources to do so outside of incredibly specific use cases
- Even then, the hardware exists in an analog world, and power surges, overheating, or even an “alpha strike” can unexpectedly change the data creating errors
 - [https://en.wikipedia.org/wiki/Alpha_strike_\(engineering\)](https://en.wikipedia.org/wiki/Alpha_strike_(engineering))
- There are also malicious attacks to consider, like Row Hammer that can affect RAM due to the physical design of the memory components
 - https://en.wikipedia.org/wiki/Row_hammer

BSIMM12

Another aspect to consider is that, in practice, functionality is given priority over security

- Which means security is “added on” at the end
- This usually gives poor results

One method to develop secure software is to utilize a model for developing a secure system

BSIMM: Building Security In Maturity Model

According to their FAQ:

What is BSIMM?

- *BSIMM (pronounced “bee simm”) is short for Building Security In Maturity Model. BSIMM is a study of real-world software security initiatives organized so that you can determine where you stand with your software security initiative and how to evolve your efforts over time.*
- <https://www.bsimm.com/about/faq.html>

BSIMM12

More from their FAQ:

Why software security?

- *Software security is about building software to be secure even when it's under attack. As we've learned from years of reviewing network security breaches, protecting software is much easier if the software is built with security in mind. Furthermore, security is a property and not a thing, so software security—being resistant to attack—involves much more than simply adding security features like encryption or passwords to software.*

History, from *BSIMM12: 2021 Insights & Trends Report*

- <https://www.bsimm.com/content/dam/bsimm/reports/bsimm12.pdf>
- *In 2008, consultant, research, and data experts from what is now the Synopsys Software Integrity Group set out to gather data on the different paths that organizations take to address the challenges of securing software. **Their goal was to examine organizations that were highly effective in software security initiatives, to conduct in-person interviews on those organizations' activities, and to publish their findings.***

The result was the Building Security In Maturity Model (better known as the BSIMM)—a descriptive model that provides a baseline of observed activities for software security initiatives. Because these initiatives often use different methodologies and different terminology, the BSIMM also creates a common vocabulary for software security initiatives.

Reference: BSIMM FAQ: <https://www.bsimm.com/about/faq.html>

BSIMM12

More from their FAQ:

Whom did you study?

- *There are 128 firms included in BSIMM12. On average, they had practiced software security for 4.4 years at the time of their current assessment (with values ranging from less than a year to 16 years as of September 2021). All 128 firms agree that the success of their initiatives hinges on having an internal group devoted to software security—a **software security group (SSG)**. **The average size of an SSG is 22.2 people (the smallest is 1, the largest is 892, and the median is 7.0)**. Often there is a satellite group of others (developers, architects, and people in the organization directly engaged in and promoting software security), and that group on average consists of 50.4 people (the smallest is 0, the largest is 1,500, and the median is 1). The average number of developers in participating organizations is 3,113.6 (the smallest is 5, the largest is 100,000, and the median is 850), yielding an average ratio of SSG to development of 2.59% (the median is 0.74%). All told, BSIMM describes the work of 9,285 SSG members and satellite staff working together to secure software that powers—nearly 153,519 applications—and is built by 398,544 developers.*

Reference: BSIMM FAQ: <https://www.bsimm.com/about/faq.html>

BSIMM12

You should be familiar with the 4 domains and 12 practices that make up BSIMM12

- <https://www.bsimm.com/content/dam/bsimm/reports/bsimm12.pdf>

THE BSIMM FRAMEWORK

BSIMM12 is organized as a set of 122 activities in a software security framework. The framework includes 12 practices that are organized into four domains, as shown in Table A.





DOMAINS			
 GOVERNANCE	 INTELLIGENCE	 SSDL TOUCHPOINTS	 DEPLOYMENT
Practices that help organize, manage, and measure a software security initiative. Staff development is also a central governance practice.	Practices that result in collections of corporate knowledge used in carrying out software security activities throughout the organization. Collections include both proactive security guidance and organizational threat modeling.	Practices associated with analysis and assurance of particular software development artifacts and processes. All software security methodologies include these practices.	Practices that interface with traditional network security and software maintenance organizations. Software configuration, maintenance, and other environment issues have direct impact on software security.
PRACTICES			
GOVERNANCE	INTELLIGENCE	SSDL TOUCHPOINTS	DEPLOYMENT
1. Strategy & Metrics (SM) 2. Compliance & Policy (CP) 3. Training (T)	4. Attack Models (AM) 5. Security Features & Design (SFD) 6. Standards & Requirements (SR)	7. Architecture Analysis (AA) 8. Code Review (CR) 9. Security Testing (ST)	10. Penetration Testing (PT) 11. Software Environment (SE) 12. Configuration Management & Vulnerability Management (CMVM)

TABLE A. THE SOFTWARE SECURITY FRAMEWORK. Twelve practices align with the four high-level domains.

BSIMM Skeleton

In a maturity model, there are levels that an organization attempts to achieve

Increasing levels indicate the use of more mature practices

- BSIMM has grouped their 12 practices into descriptions of activities that firms were conducting at the three different levels
- The following charts illustrate this for each of the domains, identifying the different activities for each of the levels for each domain

BSIMM Skeleton - Governance

THE BSIMM SKELETON

GOVERNANCE		
STRATEGY & METRICS (SM)	COMPLIANCE & POLICY (CP)	TRAINING (T)
LEVEL 1	LEVEL 1	LEVEL 1
<ul style="list-style-type: none"> [SM1.1] Publish process and evolve as necessary. [SM1.3] Educate executives on software security. [SM1.4] Implement lifecycle instrumentation and use to define governance. 	<ul style="list-style-type: none"> [CP1.1] Unify regulatory pressures. [CP1.2] Identify PII obligations. [CP1.3] Create policy. 	<ul style="list-style-type: none"> [T1.1] Conduct software security awareness training. [T1.7] Deliver on-demand individual training. [T1.8] Include security resources in onboarding.
LEVEL 2	LEVEL 2	LEVEL 2
<ul style="list-style-type: none"> [SM2.1] Publish data about software security internally and drive change. [SM2.2] Verify release conditions with measurements and track exceptions. [SM2.3] Create or grow a satellite. [SM2.6] Require security sign-off prior to software release. [SM2.7] Create evangelism role and perform internal marketing. 	<ul style="list-style-type: none"> [CP2.1] Build PII inventory. [CP2.2] Require security sign-off for compliance-related risk. [CP2.3] Implement and track controls for compliance. [CP2.4] Include software security SLAs in all vendor contracts. [CP2.5] Ensure executive awareness of compliance and privacy obligations. 	<ul style="list-style-type: none"> [T2.5] Enhance satellite through training and events. [T2.8] Create and use material specific to company history. [T2.9] Deliver role-specific advanced curriculum.
LEVEL 3	LEVEL 3	LEVEL 3
<ul style="list-style-type: none"> [SM3.1] Use an internal tracking application with portfolio view. [SM3.2] SSI efforts are part of external marketing. [SM3.3] Identify metrics and use them to drive resourcing. [SM3.4] Integrate software-defined lifecycle governance. 	<ul style="list-style-type: none"> [CP3.1] Create a regulator compliance story. [CP3.2] Impose policy on vendors. [CP3.3] Drive feedback from software lifecycle data back to policy. 	<ul style="list-style-type: none"> [T3.1] Reward progression through curriculum. [T3.2] Provide training for vendors and outsourced workers. [T3.3] Host software security events. [T3.4] Require an annual refresher. [T3.5] Establish SSG office hours. [T3.6] Identify new satellite members through observation.

TABLE B. THE BSIMM SKELETON. Within the SSF, the 122 activities are organized across different levels.

BSIMM Skeleton - Intelligence

INTELLIGENCE		
ATTACK MODELS (AM)	SECURITY FEATURES & DESIGN (SFD)	STANDARDS & REQUIREMENTS (SR)
LEVEL 1	LEVEL 1	LEVEL 1
<ul style="list-style-type: none"> [AM1.2] Create a data classification scheme and inventory. [AM1.3] Identify potential attackers. [AM1.5] Gather and use attack intelligence. 	<ul style="list-style-type: none"> [SFD1.1] Integrate and deliver security features. [SFD1.2] Engage the SSG with architecture teams. 	<ul style="list-style-type: none"> [SR1.1] Create security standards. [SR1.2] Create a security portal. [SR1.3] Translate compliance constraints to requirements.
LEVEL 2	LEVEL 2	LEVEL 2
<ul style="list-style-type: none"> [AM2.1] Build attack patterns and abuse cases tied to potential attackers. [AM2.2] Create technology-specific attack patterns. [AM2.5] Maintain and use a top <i>N</i> possible attacks list. [AM2.6] Collect and publish attack stories. [AM2.7] Build an internal forum to discuss attacks. 	<ul style="list-style-type: none"> [SFD2.1] Leverage secure-by-design components and services. [SFD2.2] Create capability to solve difficult design problems. 	<ul style="list-style-type: none"> [SR2.2] Create a standards review board. [SR2.4] Identify open source. [SR2.5] Create SLA boilerplate.
LEVEL 3	LEVEL 3	LEVEL 3
<ul style="list-style-type: none"> [AM3.1] Have a research group that develops new attack methods. [AM3.2] Create and use automation to mimic attackers. [AM3.3] Monitor automated asset creation. 	<ul style="list-style-type: none"> [SFD3.1] Form a review board or central committee to approve and maintain secure design patterns. [SFD3.2] Require use of approved security features and frameworks. [SFD3.3] Find and publish secure design patterns from the organization. 	<ul style="list-style-type: none"> [SR3.1] Control open source risk. [SR3.2] Communicate standards to vendors. [SR3.3] Use secure coding standards. [SR3.4] Create standards for technology stacks.

TABLE B. THE BSIMM SKELETON. Within the SSF, the 122 activities are organized across different levels.

BSIMM Skeleton – SSDL Touchpoints

SSDL TOUCHPOINTS		
ARCHITECTURE ANALYSIS (AA)	CODE REVIEW (CR)	SECURITY TESTING (ST)
LEVEL 1	LEVEL 1	LEVEL 1
<ul style="list-style-type: none"> [AA1.1] Perform security feature review. [AA1.2] Perform design review for high-risk applications. [AA1.3] Have SSG lead design review efforts. [AA1.4] Use a risk methodology to rank applications. 	<ul style="list-style-type: none"> [CR1.2] Perform opportunistic code review. [CR1.4] Use automated tools. [CR1.5] Make code review mandatory for all projects. [CR1.6] Use centralized reporting to close the knowledge loop. [CR1.7] Assign tool mentors. 	<ul style="list-style-type: none"> [ST1.1] Ensure QA performs edge/boundary value condition testing. [ST1.3] Drive tests with security requirements and security features. [ST1.4] Integrate opaque-box security tools into the QA process.
LEVEL 2	LEVEL 2	LEVEL 2
<ul style="list-style-type: none"> [AA2.1] Define and use AA process. [AA2.2] Standardize architectural descriptions. 	<ul style="list-style-type: none"> [CR2.6] Use automated tools with tailored rules. [CR2.7] Use a top N bugs list (real data preferred). 	<ul style="list-style-type: none"> [ST2.4] Share security results with QA. [ST2.5] Include security tests in QA automation. [ST2.6] Perform fuzz testing customized to application APIs.
LEVEL 3	LEVEL 3	LEVEL 3
<ul style="list-style-type: none"> [AA3.1] Have engineering teams lead AA process. [AA3.2] Drive analysis results into standard design patterns. [AA3.3] Make the SSG available as an AA resource or mentor. 	<ul style="list-style-type: none"> [CR3.2] Build a capability to combine assessment results. [CR3.3] Create capability to eradicate bugs. [CR3.4] Automate malicious code detection. [CR3.5] Enforce coding standards. 	<ul style="list-style-type: none"> [ST3.3] Drive tests with risk analysis results. [ST3.4] Leverage coverage analysis. [ST3.5] Begin to build and apply adversarial security tests (abuse cases). [ST3.6] Implement event-driven security testing in automation.

TABLE B. THE BSIMM SKELETON. *Within the SSF, the 122 activities are organized across different levels.*

BSIMM Skeleton - Deployment

DEPLOYMENT		
PENETRATION TESTING (PT)	SOFTWARE ENVIRONMENT (SE)	CONFIGURATION MANAGEMENT & VULNERABILITY MANAGEMENT (CMVM)
LEVEL 1	LEVEL 1	LEVEL 1
<ul style="list-style-type: none"> [PT1.1] Use external penetration testers to find problems. [PT1.2] Feed results to the defect management and mitigation system. [PT1.3] Use penetration testing tools internally. 	<ul style="list-style-type: none"> [SE1.1] Use application input monitoring. [SE1.2] Ensure host and network security basics are in place. 	<ul style="list-style-type: none"> [CMVM1.1] Create or interface with incident response. [CMVM1.2] Identify software defects found in operations monitoring and feed them back to development.
LEVEL 2	LEVEL 2	LEVEL 2
<ul style="list-style-type: none"> [PT2.2] Penetration testers use all available information. [PT2.3] Schedule periodic penetration tests for application coverage. 	<ul style="list-style-type: none"> [SE2.2] Define secure deployment parameters and configurations. [SE2.4] Protect code integrity. [SE2.5] Use application containers to support security goals. [SE2.6] Ensure cloud security basics. [SE2.7] Use orchestration for containers and virtualized environments. 	<ul style="list-style-type: none"> [CMVM2.1] Have emergency response. [CMVM2.2] Track software bugs found in operations through the fix process. [CMVM2.3] Develop an operations inventory of software delivery value streams.
LEVEL 3	LEVEL 3	LEVEL 3
<ul style="list-style-type: none"> [PT3.1] Use external penetration testers to perform deep-dive analysis. [PT3.2] Customize penetration testing tools. 	<ul style="list-style-type: none"> [SE3.2] Use code protection. [SE3.3] Use application behavior monitoring and diagnostics. [SE3.6] Enhance application inventory with operations bill of materials. 	<ul style="list-style-type: none"> [CMVM3.1] Fix all occurrences of software bugs found in operations. [CMVM3.2] Enhance the SSDL to prevent software bugs found in operations. [CMVM3.3] Simulate software crises. [CMVM3.4] Operate a bug bounty program. [CMVM3.5] Automate verification of operational infrastructure security. [CMVM3.6] Publish risk data for deployable artifacts. [CMVM3.7] Streamline incoming responsible vulnerability disclosure.

TABLE B. THE BSIMM SKELETON. Within the SSF, the 122 activities are organized across different levels.

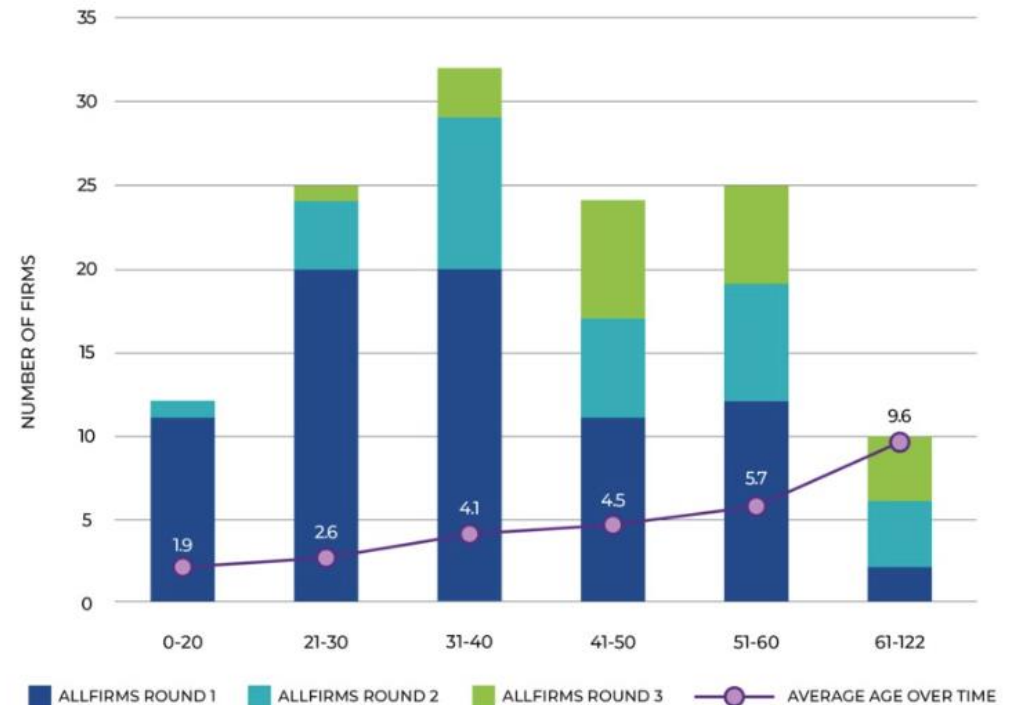
BSIMM Ratings and Improvements

From the FAQ:

Is everybody in the study equally good at software security?

- *No. By computing a score for each firm in the study, we can also take a look at relative maturity and average maturity for one firm against the others. The majority of BSIMM12 participants have a score in the 31 to 40 range, with an average SSG age of 4.1 years.*

We're pleased that BSIMM continues to grow year after year. The BSIMM project has grown from 9 participating companies in 2008 to 128 in 2021, with now nearly 3,000 software security group members and over 6,000 satellite (aka "security champions") members.



Reference: BSIMM FAQ: <https://www.bsimm.com/about/faq.html>

One Last Framework

Based on the CSF, *NISTIR 7621 Revision 1 Small Business Information Security: The Fundamentals*, modified the CSF specifically for small businesses who may not have as many resources as a larger organization

- <https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.7621r1.pdf>

You will be reviewing this document as part of Assignment 1, and it will be a good reference for your course project