OWASP Los Angeles

Security Architecture
What is it? How to deploy it?

June 28, 2023

Miguel (Mike) O. Villegas
CISO, CTO, CISSP, CISA, CDPSE, CEH,
CSX|F, CSX|A, ISO27001 Lead Implementer
mike.villegas@isecureprivacy.com
213.453.6174

iSecurePrivacy LLC

OWASP

Cybersecurity Architecture

Cybersecurity architecture is the discipline of planning out strategically the security measures of the organization.

Cybersecurity architects create a blueprint on how these security measures are effectively planned, designed, justified, tested, deployed, measured, and maintained.

There are also different kinds of architects that play a role in deploying a cybersecurity architecture. These include:

- system architects,
- network architects,
- software architects,
- application architects,
- cloud architects,
- data architects, and several others.

Architect

Architect

Noun. /’a(r)ki,tekt/

1. A professional who sleeps 2 hours a day
2. Gets excited by details no one cares about
3. Need to do precision work based on questionable client information
4. Solves a problem you did not know you had, in a way you do not understand

Architect (Really)

Architect

Noun. /’a(r)ki,tekt/

1. Design: Creates the overall concept and layout of a structure
2. Planning: Determines what is needed, what cannot be done, what is required
3. Technical: Provides guidance on construction methods, materials, techniques
4. Project Management: Is the PM or works closely with PMs to oversee implementation of design
5. Collaboration: Works closely with clients, engineers, designers, developers, and stakeholders

Cybersecurity Architect

Do you have a Cybersecurity Architect?

If you do, that is AWESOME!!!

If you do not, guess what? You ARE the Cybersecurity Architect!!

Information Security and Cybersecurity

Information Security refers to the practice of protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction. It covers all types of information, including electronic and physical, and includes policies, procedures, and controls to ensure the confidentiality, integrity, and availability of information.

Cybersecurity specifically focuses on protecting information and information systems from cyber threats, which include computer viruses, malware, phishing attacks, hacking, and other malicious activities carried out through the internet and other electronic communication channels. Cybersecurity involves protecting networks, devices, and data from unauthorized access, modification, or destruction.

Define Cybersecurity Architecture

Cybersecurity architecture refers to the design and implementation of a comprehensive security framework that protects an organization's networks, systems, and data from cyber threats.

Cybersecurity architecture typically includes several layers of security controls that work together to create a defense-in-depth approach to security.

Approaches to Security

- **Security through Obscurity** – if no one knows how to use the system, no one will know how to hack it
- **Security through Obsolesce** – using very old antiquated products in the hopes that no one knows what to do with it.
- **Security through Minority** – use the least number of products such that those with the skills required to use them are low.
- **Security through Diversity** – use as many systems as possible so that no one person knows all the systems used.
- **Security by Design** – the best approach to Security; consider security in all phases.

Approaches to Security

IGNORANCE IS NOT A CONTROL

3 Principles of Architecture Design

Over the years, I have determined there are three basic principles of an architecture (or any system) design:

Villegas	OWASP SAMM
Comprehensive	Measurable
Flexible	Actionable
Easy to Use	Versatile

These almost sound oxymoronic and difficult to achieve but if well thought out, in aggregate they are achievable.

5W1H Problem Solving Approach

5W1H		
What?		Business, Industry, Scope, Architecture
Where?		Legacy, Cloud, Outsourced
Why?		Growth, Compliance, Client-Driven, Targets
Who?		Internal, Customers, Vendors, Partners, Biz Culture
When?		1-3-5 Year Plan, Aligned with Biz Strategy, Yesterday
How?		CSP SaaS, IaaS, PaaS; Dev Internally, Outsourced, MSP

As you review, design, deploy, or audit a security architecture, you need to ask these six questions at every step.

Security Architecture Frameworks and Security Frameworks

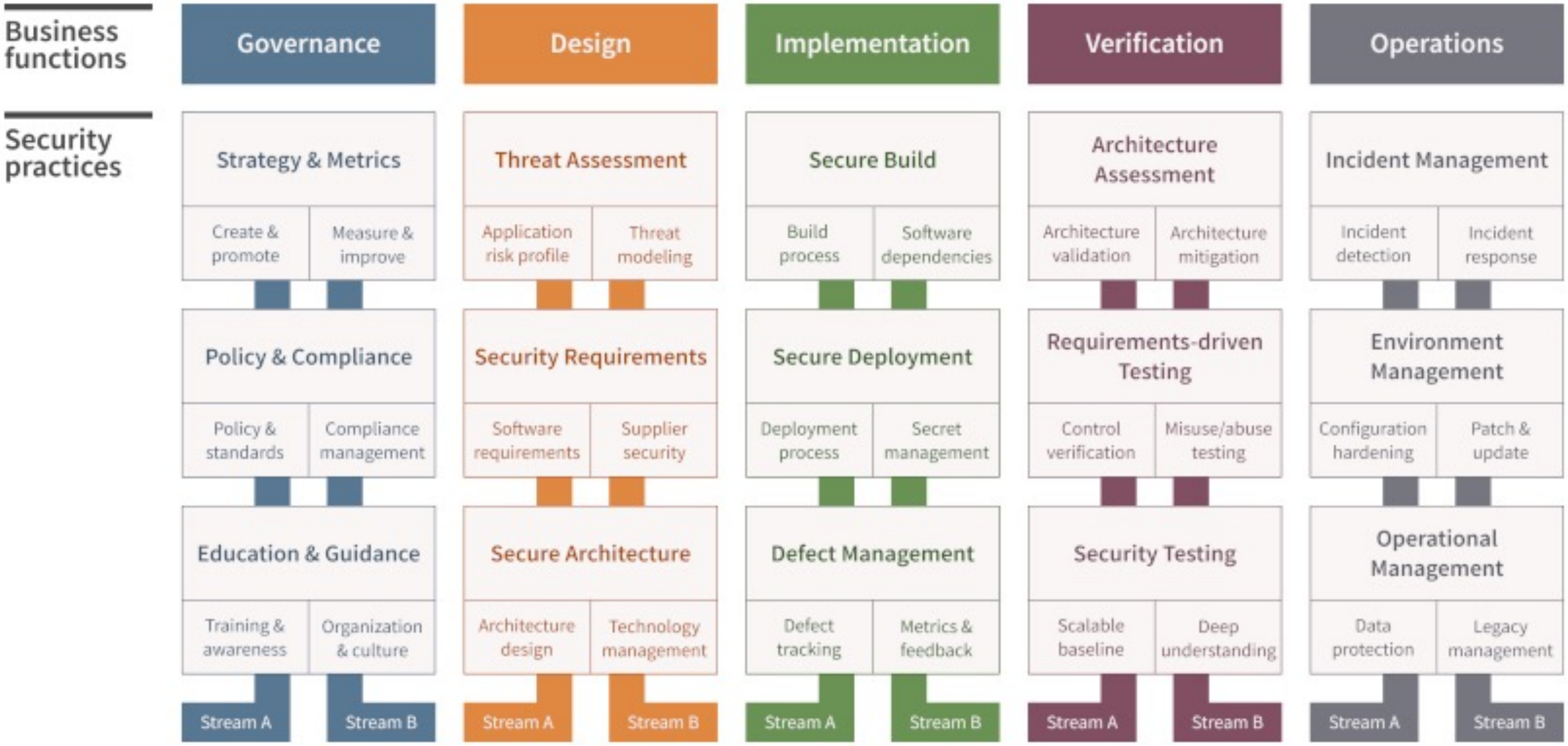
Security Architectures

- ***SABSA (Sherwood Applied Business Security Architecture)***
- *CISSP – ISSAP (Information Systems Security Architecture Professional) – 6 domains*
  1. Access Controls Systems & Methodology
  2. Communications & Network Security
  3. Cryptography
  4. Security Architecture Analysis
  5. Technology Related BCP & DRP
  6. Physical Security Considerations

Security Frameworks

- ***OWASP SAMM (Software Assurance Maturity Model)***
- ***PCI Software Security Framework (SSF)***
- *NIST Cybersecurity Framework (CSF)*
- *NIST SP 800-160* Engineering Trustworthy Secure Systems
- *AWS Security by Design Framework*
- *COBIT by ISACA (7 Domains)*
  1. Audit & Assurance
  2. Emerging Technology
  3. Governance
  4. Information Security
  5. Information Technology
  6. Privacy
  7. Risk
- *Microsoft Cybersecurity Reference Architecture (MCRA)*

OWASP SAMM v2.0



Business functions	Governance		Design		Implementation		Verification		Operations	
Security practices	Strategy & Metrics		Threat Assessment		Secure Build		Architecture Assessment		Incident Management	
	Create & promote	Measure & improve	Application risk profile	Threat modeling	Build process	Software dependencies	Architecture validation	Architecture mitigation	Incident detection	Incident response
	Policy & Compliance		Security Requirements		Secure Deployment		Requirements-driven Testing		Environment Management	
	Policy & standards	Compliance management	Software requirements	Supplier security	Deployment process	Secret management	Control verification	Misuse/abuse testing	Configuration hardening	Patch & update
	Education & Guidance		Secure Architecture		Defect Management		Security Testing		Operational Management	
	Training & awareness	Organization & culture	Architecture design	Technology management	Defect tracking	Metrics & feedback	Scalable baseline	Deep understanding	Data protection	Legacy management
	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B

iSecurePrivacy LLC

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

Scope

In-Scope includes

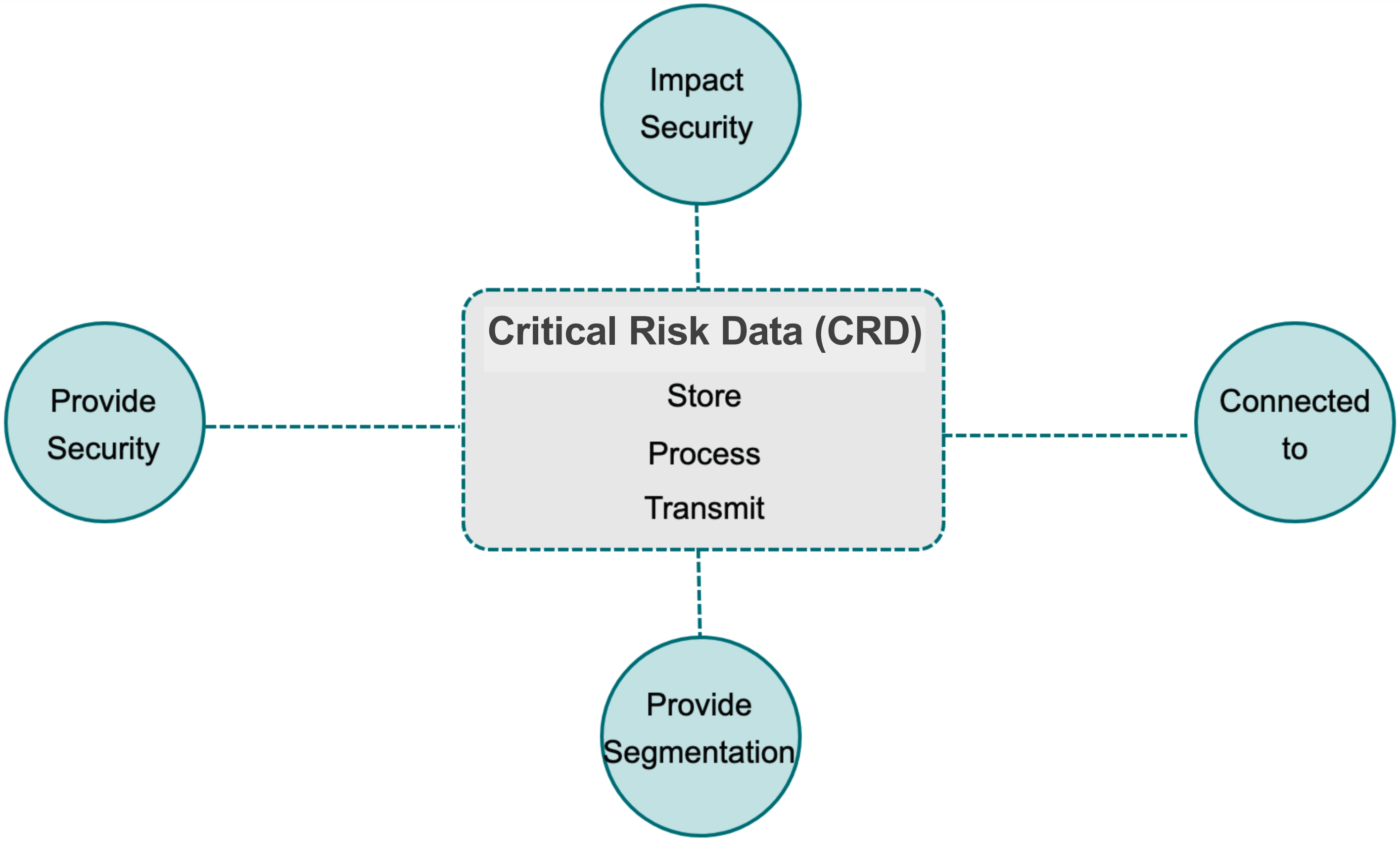
- devices (technology)
- processes
- and people

Factors for Scope:

- Tier 1 – used to store, process, or traverses through devices, processes, or people
- Tier 2 – connected to, or has an impact on security of Tier 1

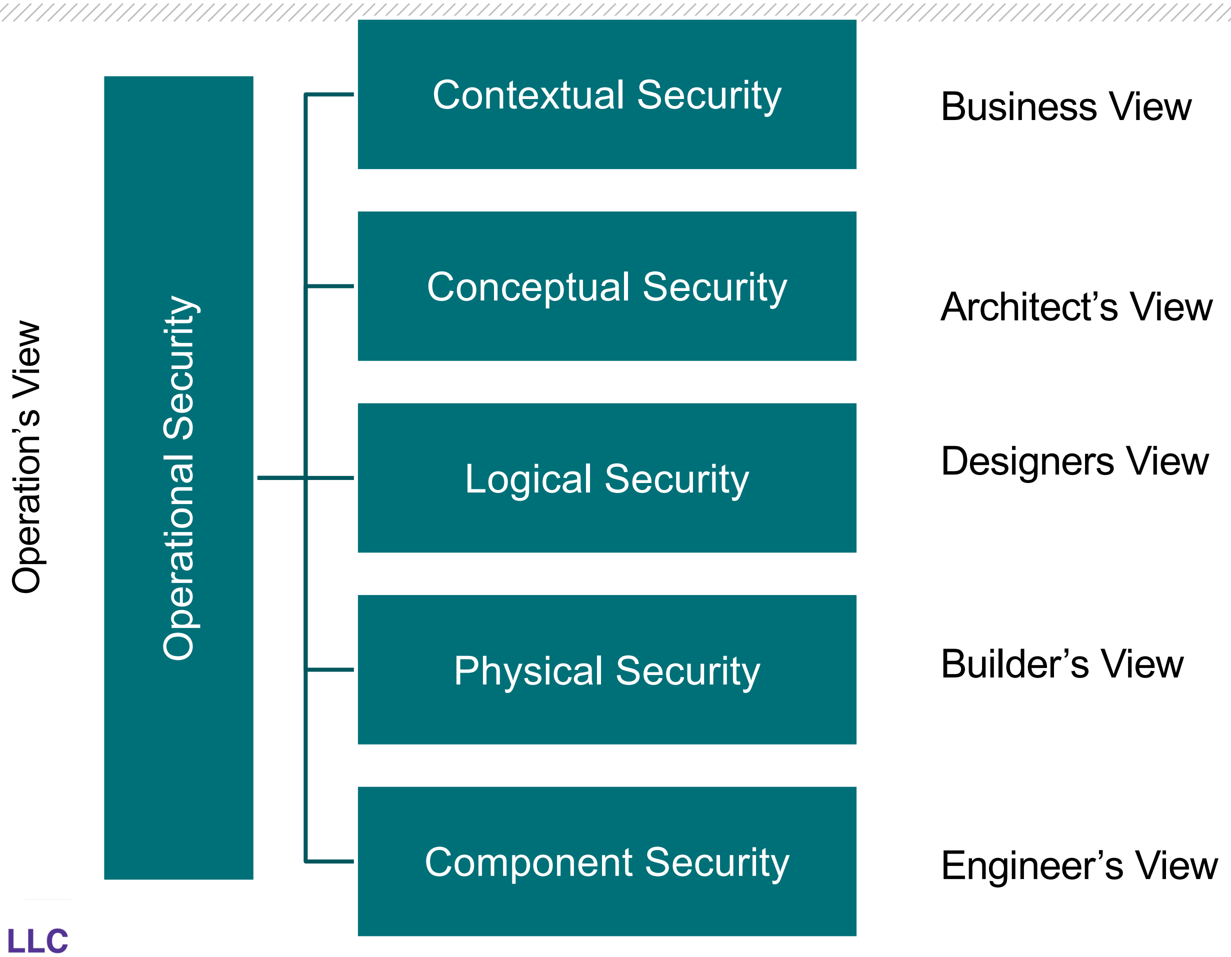
For example: Active Directory, NTP, SIEM, FIM, DNS

Scope



Impact Security
Provide Security
Critical Risk Data (CRD)
Store
Process
Transmit
Connected to
Provide Segmentation

SABSA Model for Security Architecture Development



iSecurePrivacy LLC

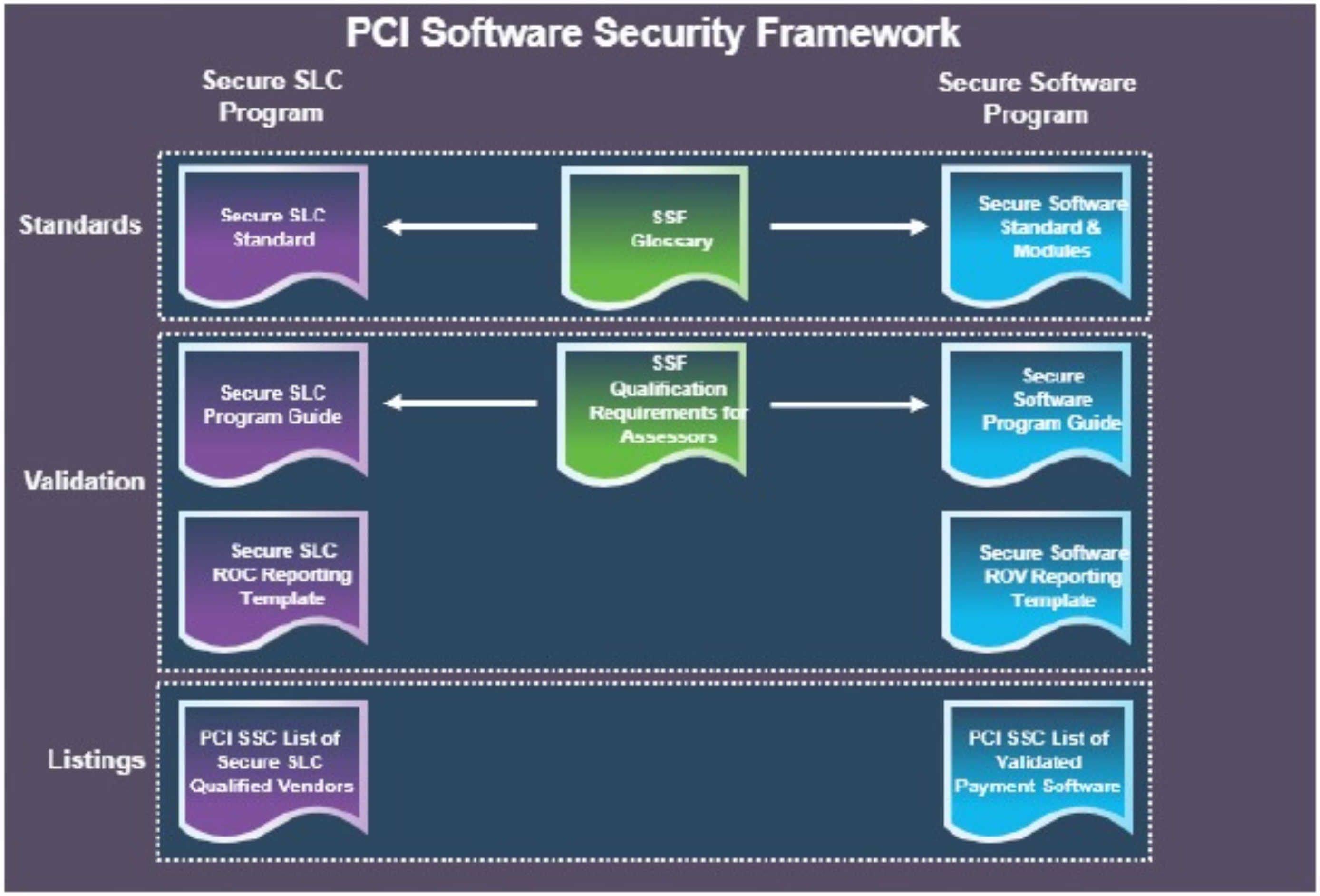
Source: Sherwood, Nicholas A. Enterprise Security Architecture: A Business-Driven Approach (p. 2). CRC Press. Kindle Edition.

36-Cell SABSA Matrix

	Assets (What)	Motivation (Why)	Process (How)	People (Who)	Location (Where)	Time (When)
Contextual	The Business	Business Risk Model	Business Process Model	Business Organization and Relationships	Business Geography	Business Time Dependencies
Conceptual	Business Attributes Profile	Control Objectives	Security Strategy and Architectural Layering	Security Entity Model and Trust Framework	Security Domain Model	Security-Related Lifetimes and Deadlines
Logical	Business Information Model	Security Policies	Security Services	Entity Schemas and Privilege Profiles	Security Domain Definitions and Associations	Security Processing Cycle
Physical	Business Data Model	Security Rules, Practices and Procedures	Security Mechanisms	Users, Applications and User Interfaces	Platform and Network Infrastructure	Control Structure Execution
Component	Detailed Data Structures	Security Standards	Security Products and Tools	Identities, Functions, Actions, and ACLs	Processes, Modes, Addresses and Protocols	Security Step Timing and Sequencing
Operational	Assurance of Operational Continuity	Operational Risk Management	Security Service Management and Support	Application and User Management Support	Security of Sites, Networks and Platforms	Security Operations Schedule

Source: Sherwood, Nicholas A. Enterprise Security Architecture: A Business-Driven Approach (p. 2). CRC Press. Kindle Edition.

PCI Secure Software Framework (SSF)



PCI Software Security Framework

Secure SLC Program

Secure Software Program

Standards

Secure SLC Standard

SSF Glossary

Secure Software Standard & Modules

Validation

Secure SLC Program Guide

SSF Qualification Requirements for Assessors

Secure Software Program Guide

Secure SLC ROC Reporting Template

Secure Software ROV Reporting Template

Listings

PCI SSC List of Secure SLC Qualified Vendors

PCI SSC List of Validated Payment Software

PCI Secure Software Framework (SSF)

Secure SLC Control Objectives	Secure Software Control Objectives
CO1: Security Responsibilities and Resources	CO1: Critical Asset Identification
CO2: Software Security Policy & Strategy	CO2: Secure Defaults
CO3: Threat Identification & Mitigation	CO3: Sensitive Data Retention
CO4: Vulnerability Detection & Mitigation	CO4: Critical Asset Protection
CO5: Change Management	CO5: Authentication & Access Control
CO6: Software Integrity Protection	CO6: Sensitive Data Protection
CO7: Sensitive Data Protection	CO7: Use of Cryptography
CO8: Vendor Security Guidance	CO8: Activity Tracking
CO9: Stakeholder Communications	CO9: Attack Detection
CO10: Software Update Information	CO10: Threat & Vulnerability Management
	CO11: Secure Software Updates
	CO12: Software Vendor Implementation Guidance

PCI Secure Software Assessment Modules

Module A – Account Data Protection Requirements
COA.1: Sensitive Authentication Data
COA.2: Cardholder Data Protection
Module B – Terminal Software Requirements
COB.1: Terminal Software Documentation
COB.2: Terminal Software Design
COB.3: Terminal Software Security Testing
COB.4: Terminal Software Security Testing
COB.5: Terminal Software Implementation Guidance
Module C – Web Software Requirements
COC.1: Web Software Components & Services
COC.2: Web Software Access Controls
COC.3: Web Software Attack Mitigation
COC.4: Web Software Communications

PCI SSF Modules

Module A deals with security requirements for software that stores, processes, or transmits account data

Module B deals with security requirements for software intended for deployment and execution on PCI-approved POI devices. Point of Interchange (POI) would be for example point-of-sale (POS) devices.

Module C deals with security requirements for payment software that uses Internet technologies, protocols, and languages to initiate or support electronic payment transactions. This includes both:

- Traditional (monolithic) web payment applications
- Cloud-native payment applications

PCI Account Data

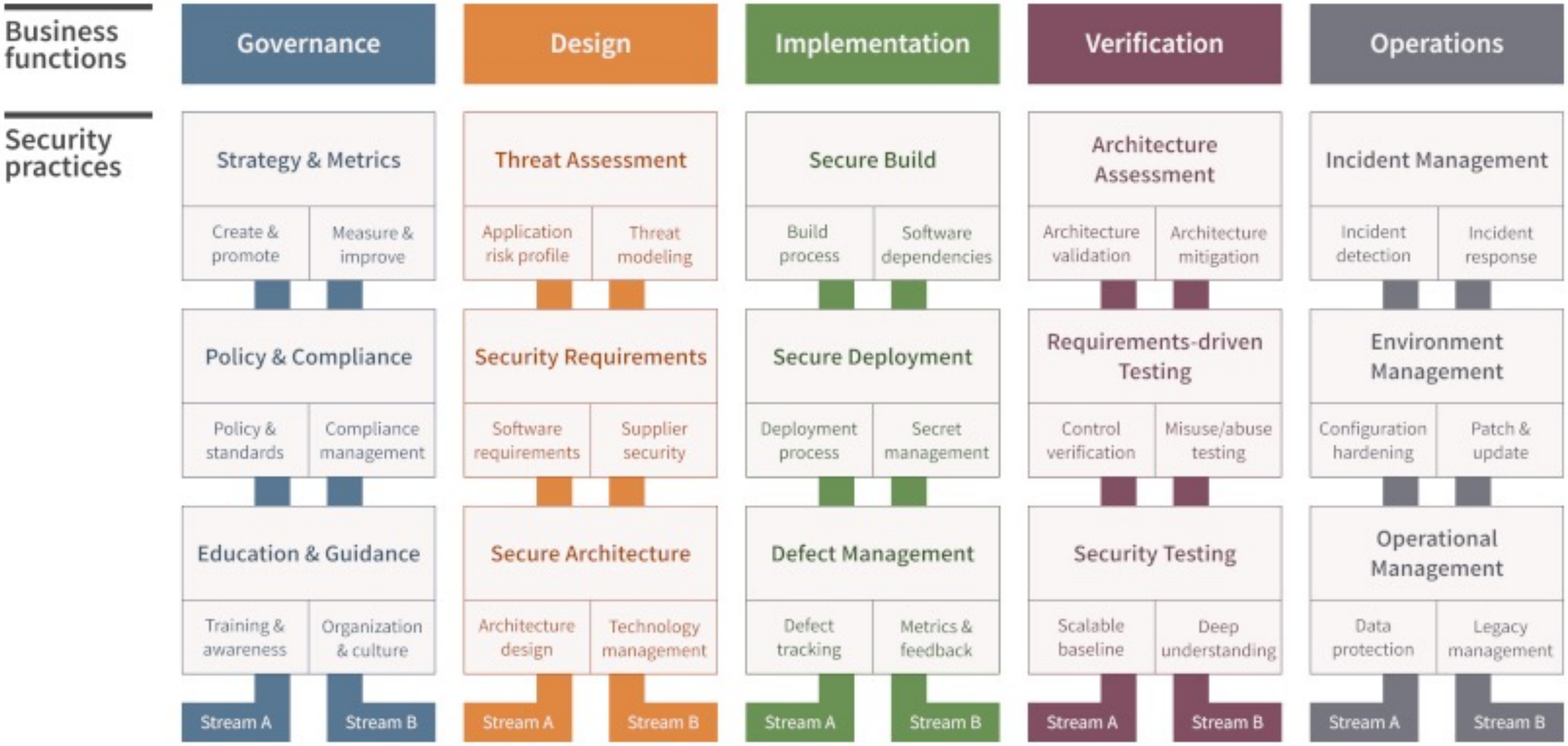
Account Data	
Cardholder Data includes:	Sensitive Authentication Data includes:
• Primary Account Number (PAN)	• Full track data (magnetic-stripe data or equivalent on a chip)
• Cardholder Name	• CAV2/CVC2/CVV2/CID
• Expiration Date	• PINs/PIN blocks
• Service Code	

PAN Obfuscation	
• Encryption	Implies it can be decrypted
• Truncation	1234-56**-****-7890 (first 6; last 4)
• Hashing	One way hashing with salt; need strong algorithm not subject to rainbow
• Tokenization	Data replaced by surrogate, proxy values or tokens
• Masking	Data displayed in viewing or entering (first 6; last 4)

PCI Account Data

Encryption	
• Storage	Hard disks, tape drives, databases, removables (e.g., thumb drives, DVD)
• Network	TLS 1.2+; VPN; MFA
• End-to-End	Encryption by default over the entire lifecycle; difficult since it relies on trust relationships between systems
• Point-to-Point	PCI encryption standard – PAN encrypted with an approved secure cryptographic device (SCD) like POS and not decrypted at all at merchant
• Application Layer	Layer 7 (human interaction layer); 5W1H controls
• Full-Disk Encryption	FDE protects from a disk being used in an unauthorized environment, but file, field, database, and selective data encryption still needed
• Hardware Security Module	HSM are hardened, tamper-resistant hardware devices to strengthen encryption of data and/or key generation and key management

OWASP SAMM v2.0



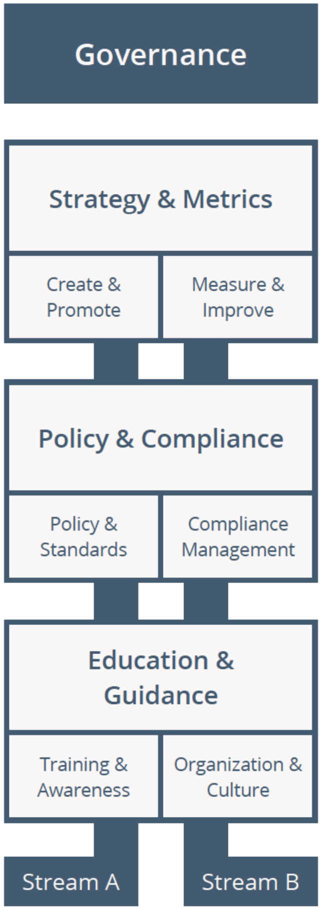
Business functions	Governance	Design	Implementation	Verification	Operations
Security practices	Strategy & Metrics	Threat Assessment	Secure Build	Architecture Assessment	Incident Management
	Create & promote Measure & improve	Application risk profile Threat modeling	Build process Software dependencies	Architecture validation Architecture mitigation	Incident detection Incident response
	Policy & Compliance	Security Requirements	Secure Deployment	Requirements-driven Testing	Environment Management
	Policy & standards Compliance management	Software requirements Supplier security	Deployment process Secret management	Control verification Misuse/abuse testing	Configuration hardening Patch & update
	Education & Guidance	Secure Architecture	Defect Management	Security Testing	Operational Management
	Training & awareness Organization & culture	Architecture design Technology management	Defect tracking Metrics & feedback	Scalable baseline Deep understanding	Data protection Legacy management
	Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B	Stream A Stream B

iSecurePrivacy LLC

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

OWASP

SAMM - Governance



Governance

Strategy & Metrics

Create & Promote | Measure & Improve

Policy & Compliance

Policy & Standards | Compliance Management

Education & Guidance

Training & Awareness | Organization & Culture

Stream A | Stream B

Stream A Create & Promote	Stream B Measure & Improve
Maturity level 1 ●○○ Identify objectives and means of measuring effectiveness of the security program.	
Identify organization drivers as they relate to the organization's risk tolerance.	Define metrics with insight into the effectiveness and efficiency of the Application Security Program.
Maturity level 2 ●●○ Establish a unified strategic roadmap for software security within the organization.	
Publish a unified strategy for application security.	Set targets and KPI's for measuring the program effectiveness.
Maturity level 3 ●●● Align security efforts with the relevant organizational indicators and asset values.	
Align the application security program to support the organization's growth.	Influence the strategy based on the metrics and organizational needs.

SAMM - Governance

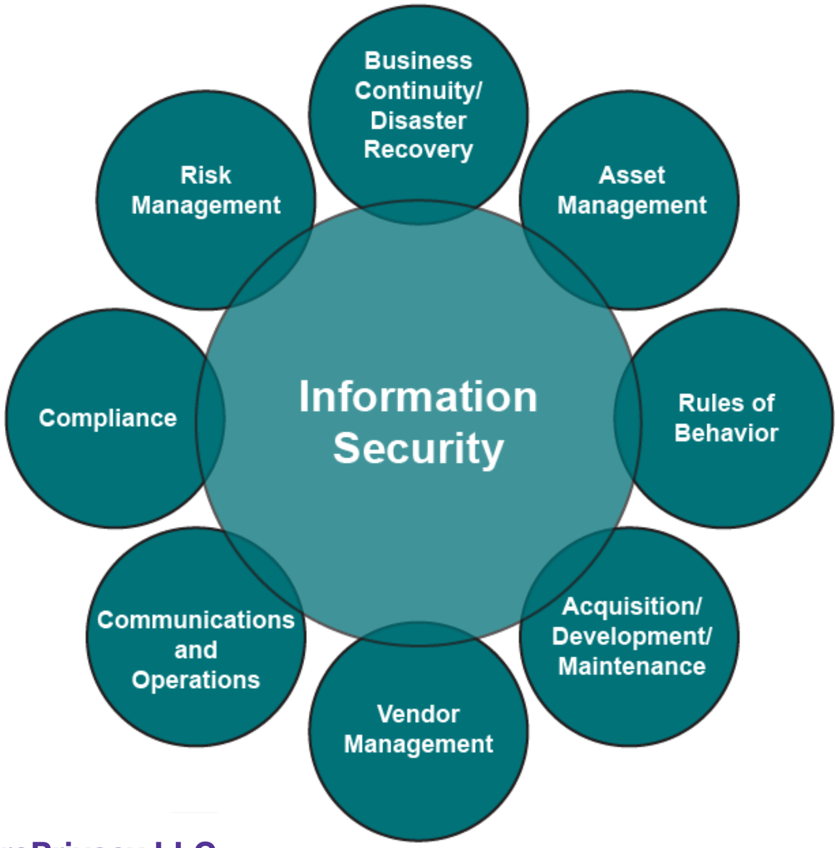
Governance – how an organization manages and measures overall software development activities. It includes

- Policies
- Procedures
- Compliance to laws
- Regulations
- Internal PnP (polices and procedures)
- Training
- Awareness

Document Types

Type	Description
Policies	Communicate required and prohibited activities and behaviors
Standards	Interpret policies in specific situations
Procedures	Provide details on how to comply with policies and standards
Guidelines	Provide general guidance on issues such as “what to do in particular circumstances.” These are not requirements to be met, but are strongly recommended.

COBIT 5 Information Security Policy Set

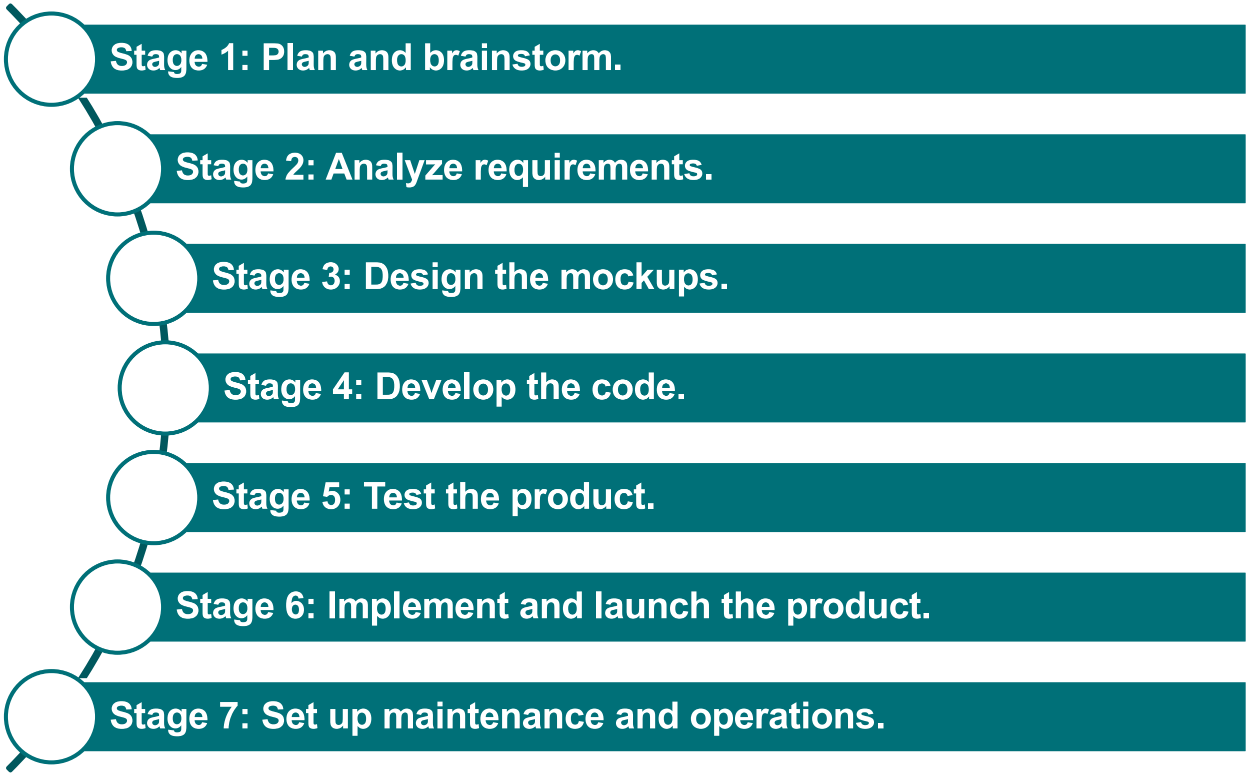

Business Continuity/ Disaster Recovery
Risk Management
Asset Management
Compliance
Information Security
Rules of Behavior
Communications and Operations
Acquisition/ Development/ Maintenance
Vendor Management

Other Info Sec Related Policies

- Access Control Policy
- Acceptable Use Policy
- Data Retention Policy
- Personnel Information Security Policy
- Social Media Policy
- Wireless Policy
- Privacy Policy
- Security Incident Response Policy
- Security Awareness Policy
- Key Management Policy
- Remote Access Policy
- Network Security Configuration Policy
- Server Security Configuration Policy
- Secure Coding Standards
- Change Control Policy
- Physical Security Policy
- Wireless Policy
- Many more…

Software Development Life Cycle (SDLC)

If you have 1 hour to chop down a tree, you should spend 45 minutes sharpening your ax



Stage 1: Plan and brainstorm.

Stage 2: Analyze requirements.

Stage 3: Design the mockups.

Stage 4: Develop the code.

Stage 5: Test the product.

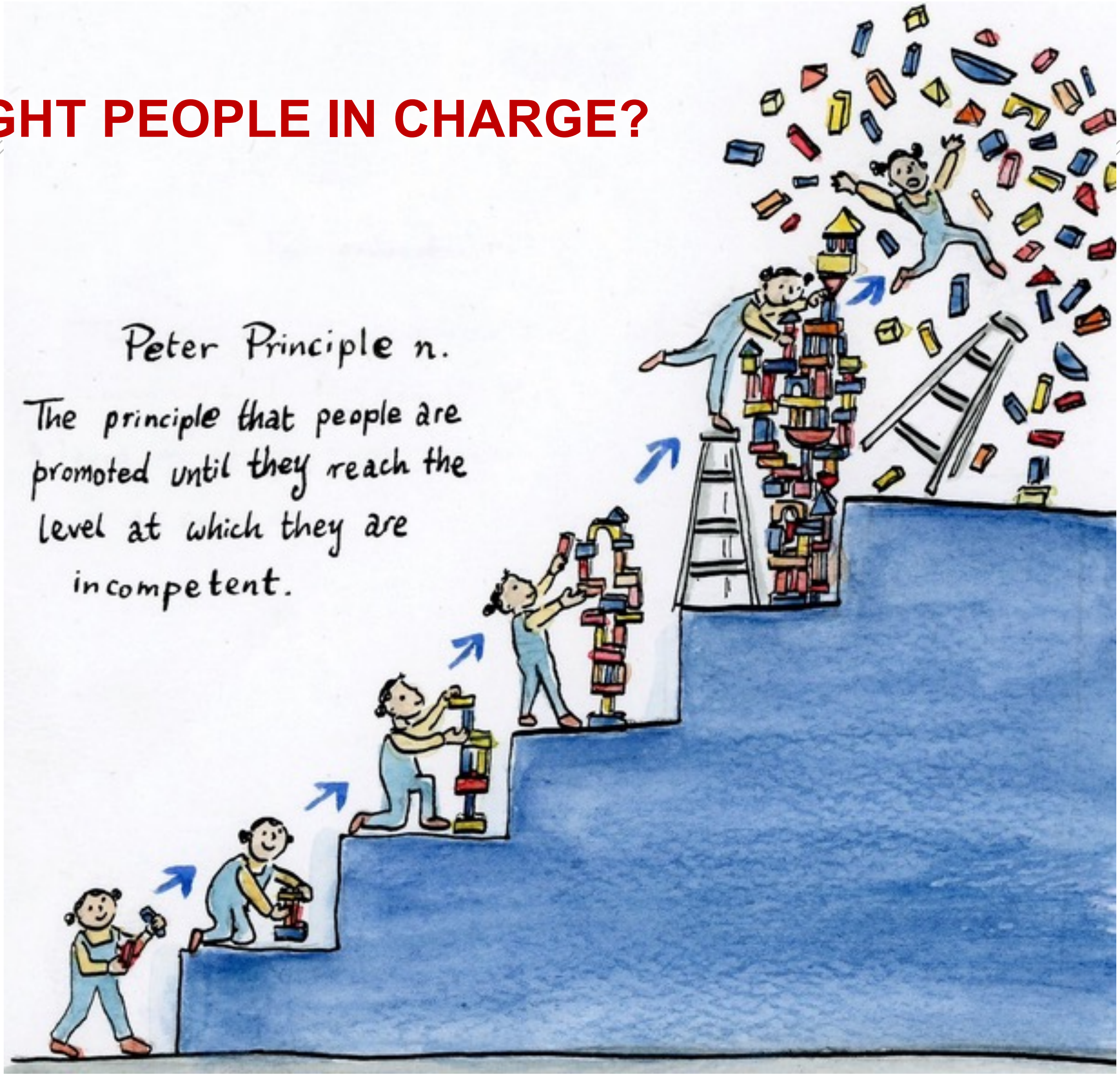
Stage 6: Implement and launch the product.

Stage 7: Set up maintenance and operations.

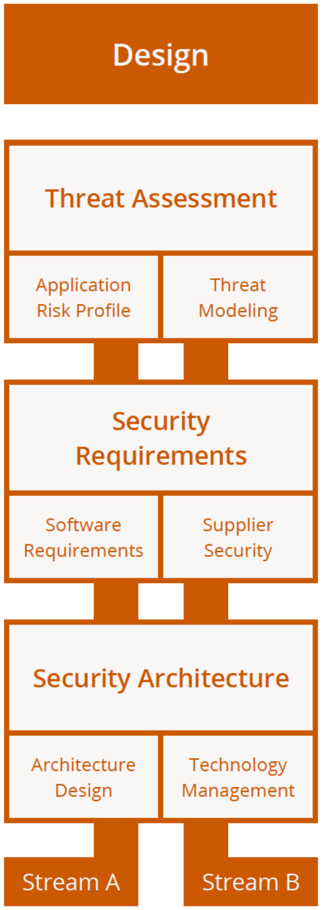
Contextual Security Architecture (Business View)

What is the business model and industry?	What are the business communication requirements?
How is cybersecurity valued by business?	Info Sec has no intrinsic value on its own
Info Sec should be a business enabler	Does the business have a documented Strategic Business Plan?
Does the business need online access, memo posted, or legacy data to run their business?	Does Info Sec have a documented Security Strategic Plan?
Is remote access a requirement?	Are these two strategic plans aligned?
Is the business dependent on successful management of the supply chain?	Does the business work with and share PII, PHI, PCI, M&A data with other entities (internal, external, government)?
Is the Internet and web frequently used for research and information gathering?	How important are: customer service, market reputation, management control, profit, regulatory compliance?
Is the business dependent on authenticated service providers for source of information?	What are the applications that drive the business and their related security needs?

ARE THE RIGHT PEOPLE IN CHARGE?


Peter Principle n.
The principle that people are promoted until they reach the level at which they are incompetent.

SAMM - Design



Design

Threat Assessment

Application Risk Profile

Threat Modeling

Security Requirements

Software Requirements

Supplier Security

Security Architecture

Architecture Design

Technology Management

Stream A

Stream B

Stream A **Application Risk Profile**	Stream B **Threat Modeling**
Maturity level 1 ●○○ Best-effort identification of high-level threats to the organization and individual projects.	
A basic assessment of the application risk is performed to understand likelihood and impact of an attack.	Perform best-effort, risk-based threat modeling using brainstorming and existing diagrams with simple threat checklists.
Maturity level 2 ●●○ Standardization and enterprise-wide analysis of software-related threats within the organization.	
Understand the risk for all applications in the organization by centralizing the risk profile inventory for stakeholders.	Standardize threat modeling training, processes, and tools to scale across the organization.
Maturity level 3 ●●● Proactive improvement of threat coverage throughout the organization.	
Periodically review application risk profiles at regular intervals to ensure accuracy and reflect current state.	Continuously optimization and automation of your threat modeling methodology.

iSecurePrivacy LLC

OWASP

SAMM - Design

Design – how an organization defines goals and creates software within development projects.

- How will the development project deal with real threats in applications?

- What are the security requirements for the software and software suppliers (vendors or internal), for example PCI SSF?

- What are the architectural risks?
  - security principles include defense in depth,
  - securing the weakest link,
  - use of secure defaults,
  - simplicity in design of security functionality,
  - secure failure,
  - balance of security and usability,
  - running with least privilege,
  - avoidance of security by obscurity - if no one knows how to use the system, no one will know how to hack it

Conceptual Security Architecture (Architect’s View)

Supported Technologies: AI, satellite, RFID, HSM, FIM, SIEM, CSP, SOC, MSP, MSSP, etc.	Obtain a comprehensive physical and logical network diagram.
Define the Business Attributes	Obtain data flow diagrams for each critical business process, including IT.
Obtain a full inventory of the following assets:	Define overall control objectives for each critical asset.
• Critical risk data	Develop a multi-layered security model
• Business and critical applications	Design security for each asset using 5W1H
• Complete list of hardware devices: layer-3, servers, mobile, end-points, etc..	Identify all security services for each critical asset: IAAA.
• List of users, business groups, IT support, Information Security	I – Identification; A – Authentication; A – Authentication; A – Audit (logging and monitoring)
• MSP, CSP, MSSP and any other service provider(s)	Design CIA into the Security Architecture Model
• Perform an Enterprise Risk Assessment	Use Multi-Factor Authentication where needed

Goal is to design the forest and not the trees.

Sensitive Data

PCI SSF CO1.1 All sensitive data stored, processed, or transmitted by the software is identified.

- all payment data (PII, ePHI, PCI, critical risk data)
- authentication credentials (passwords, tokens, certificates, DEKs, KEKs)
- cryptographic keys and related data (such as IVs and seed data for random number generators);
- system configuration data such as:
  - registry entries,
  - platform environment variables,
  - prompts for plaintext data in software allowing for the entry of PIN data, or configuration scripts.
- Where is it stored
  - Temporary storage (volatile memory)
  - Semi-permanent storage (RAM)
  - Non-volatile storage (flash drives and flash storage media)
- How is it protected
  - Cryptography
  - ACL.
  - Protected memory.
  - HSM

Web Software Components and Services

Modern software is rarely created entirely in-house and is typically composed of various bespoke code segments that are integrated with numerous components. The following software components need to be identified, verified, and tested for vulnerabilities:

- All proprietary software libraries, packages, modules, and/or code packaged in a manner that enables them to be tracked as a freestanding unit of software.
- All third-party and open-source frameworks, libraries, and code embedded in or used by the software during operation.
- All third-party software dependencies, APIs, and services called by the software during operation.

NIST refers to “provenance data” as information of the above components and services, versions, and any third-party code that may be embedded in these components.

The following shows how this information should be structured:

- CycloneDX
- SPDX.
- SWID

The software does not disclose sensitive data through unintended channels.

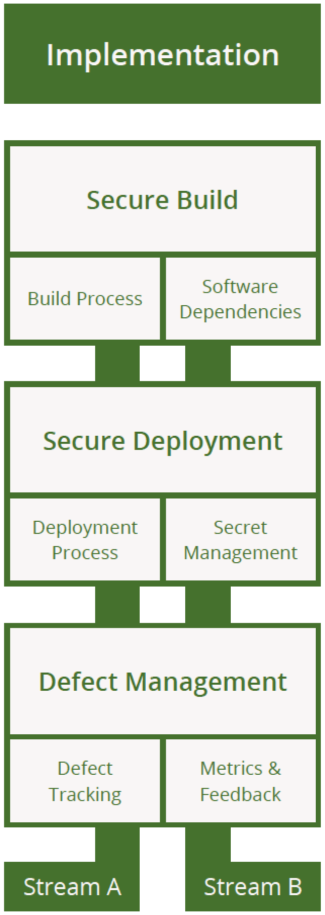
- Error messages, error logs, or memory dumps.
- Execution environments that may be vulnerable to remote side-channel attacks to expose sensitive data
- Automatic storage or exposure of sensitive data by the underlying execution environment, such as through swap-files, system error logging, keyboard spelling, and auto-correct features
- Sensors or services provided by the execution environment that may be used to extract or leak sensitive data such as through use of an accelerometer to capture input of a passphrase to be used as a seed for a cryptographic key, or through capture of sensitive data through use of cameras, near-field communication (NFC) interfaces

IF YOU DO NOT NEED IT

DO NOT STORE IT

AND SECURELY WIPE

SAMM - Implementation



Stream A **Build Process**	Stream B **Software Dependencies**

Maturity level 1 ●○○

Build process is repeatable and consistent.

Create a formal definition of the build process so that it becomes consistent and repeatable.	Create records with Bill of Materials of your applications and opportunistically analyze these.

Maturity level 2 ●●○

Build process is optimized and fully integrated into the workflow.

Automate your build pipeline and secure the used tooling. Add security checks in the build pipeline.	Evaluate used dependencies and ensure timely reaction to situations posing risk to your applications.

Maturity level 3 ●●●

Build process helps prevent known defects from entering the production environment.

Define mandatory security checks in the build process and ensure that building non-compliant artifacts fails.	Analyze used dependencies for security issues in a comparable way to your own code.

SAMM - Implementation

Implementation – how an organization builds and deploys software components and related defects.

- What external libraries does it use and how secure are they, such as open source?
- Do you use automated security checks (such as DAST/SAST tools) in the pipeline to development?
- How do you handle code signing?

SAMM - Implementation

SAST Tools (Sample)

1. Synopsys (Coverity)
2. Checkmarx
3. Veracode
4. SonarQube
5. Micro Focus (Fortify Static Code Analyzer)
6. Klocwork

DAST Tools (Sample)

1. Rapid7 (AppSpider)
2. Acunetix
3. Micro Focus (Fortify WebInspect)
4. OWASP ZAP
5. Trustwave (App Scanner)
6. PortSwigger (Burp Suite)

Logical Security Architecture (Designer’s View)

Flesh out the bones of the Conceptual Framework	Define interdependencies between systems and applications
Develop information security policies, procedures, and standards	Define access to assets based on Principle of Least Privilege and RBAC
Design security services into these architectures:	Identify solutions (products, services, and processes) for IAAA for each critical asset
• Certificate management	Perform a Gap Assessment, controls not in place:
• Directory services	• Inherent limitations
• Access control	• Lack of resources (time, people, funds)
• Entity authentication	• Lack of knowledge
• Service management	• Lack of management support
• Incident response	Perform Privacy Impact Assessment
Perform compliance reviews (SOC2, HIPAA, ISO 27001, NY DFS, NIST SP 800 53/171 etc.)	Perform Business Impact Assessment

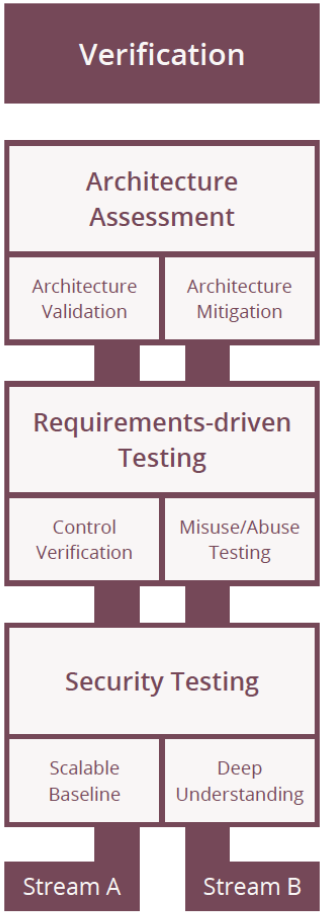
Multifactor Authentication

MFA (Multifactor Authentication) or Two-Factor Authentication should be used to strengthen authentication controls for remote access. This currently is a requirement in some regulatory and payment system security requirements.

Authentication Factors
- ❖ Something I know
- ❖ Something I have
- ❖ Something I am

Logging on twice with an ID and password is NOT MFA

SAMM - Verification



Verification

Architecture Assessment

Architecture Validation

Architecture Mitigation

Requirements-driven Testing

Control Verification

Misuse/Abuse Testing

Security Testing

Scalable Baseline

Deep Understanding

Stream A

Stream B

<table>
<tr><th>Stream A
Architecture Validation</th><th>Stream B
Architecture Mitigation</th></tr>
<tr><td colspan="2">Maturity level 1 ●○○
Review the architecture to ensure baseline mitigations are in place for typical risks.</td></tr>
<tr><td>Identify application and infrastructure architecture components and review for basic security provisioning.</td><td>Ad-hoc review of the architecture for unmitigated security threats.</td></tr>
<tr><td colspan="2">Maturity level 2 ●●○
Review the complete provision of security mechanisms in the architecture.</td></tr>
<tr><td>Validate the architecture security mechanisms.</td><td>Analyze the architecture for known threats.</td></tr>
<tr><td colspan="2">Maturity level 3 ●●●
Review the architecture effectiveness and feedback results to improve the security architecture.</td></tr>
<tr><td>Review of the architecture components' effectiveness.</td><td>Feed the architecture review results back into the enterprise architecture, organization design principles and patterns, security solutions and reference architectures.</td></tr>
</table>

SAMM - Verification

Verification – this involves quality assurance work such as testing, compliance, documentation, evaluation activities, and approvals. Verification of:

- Architecture - Review the effectiveness of the architecture components and their provided security mechanisms in terms of alignment with the overall strategy of the organization, and scrutinize the degree of availability, scalability and enterprise readiness of the chosen security solutions.
- Requirements driven testing –
  - positive testing (control verification) which validates security controls are being met and prevents bugs being introduced into the application environment through regression testing, and
  - negative testing (misuse/abuse) which includes **fuzzing** (consists in finding implementation bugs using malformed/semi-malformed data injection in an automated fashion; inserting arbitrary code or wrong data types), misuse/abuse cases (consists of logical attacks that ensures nothing happens that should not be allowed; for example, reasonable checks, DOS attacks, business logic testing) https://github.com/samhocevar/zzuf/releases
- Security testing – baseline testing using Static Applications Security Testing (SAST), Dynamic Application Security Testing (DAST) or Interactive Application Security Testing (IAST) which is a hybrid of both SAST and DAST that combines the strength of both.

Controls to Mitigate Software Attacks

Examples of software security controls include:

- Input and output validation
- Authentication and Password Management
- Parameterization
- Escaping
- Segmentation
- Logging
- Defaults modified
- Encryption
- Detection of anomalous behavior

SECURE CODING STANDARDS

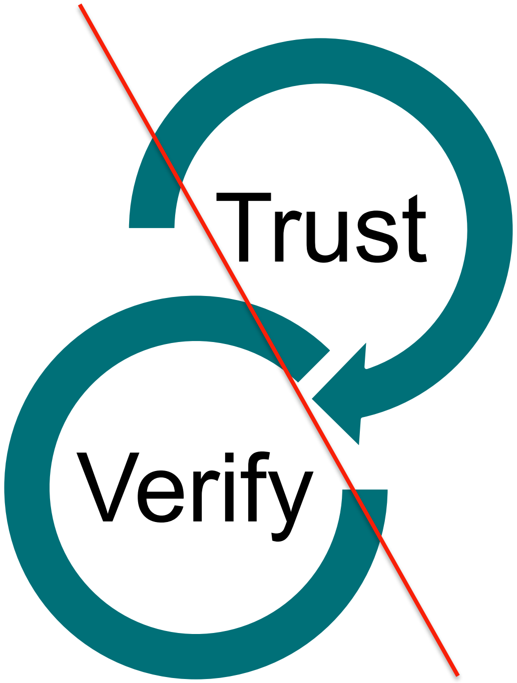
Input Validation Guidelines (sample)

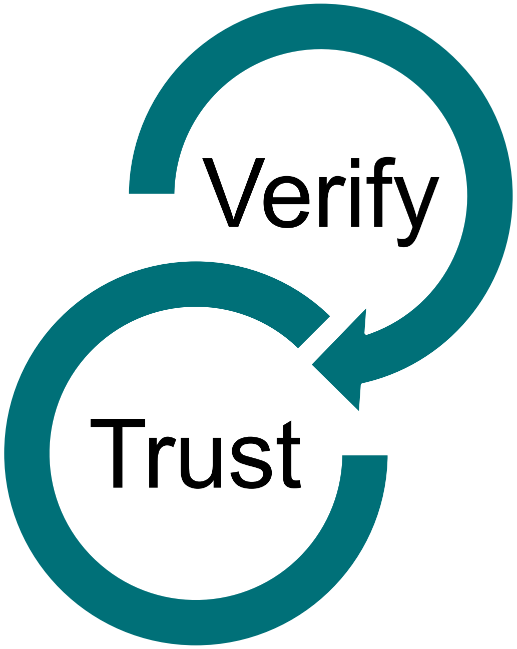
- Conduct all data validation on a trusted system (e.g., The server)
- Identify all data sources and classify them into trusted and untrusted. Validate all data from untrusted sources (e.g., Databases, file streams, etc.)
- There should be a centralized input validation routine for the application
- Specify proper character sets, such as UTF-8, for all sources of input
- Encode data to a common character set before validating (Canonicalize)
- All validation failures should result in input rejection
- Determine if the system supports UTF-8 extended character sets and if so, validate after UTF-8 decoding is completed
- Validate all client provided data before processing, including all parameters, URLs and HTTP header content (e.g. Cookie names and values). Be sure to include automated post backs from JavaScript, Flash or other embedded code
- Verify that header values in both requests and responses contain only ASCII characters
- Validate data from redirects (An attacker may submit malicious content directly to the target of the redirect, thus circumventing application logic and any validation performed before the redirect)
- Validate for expected data types
- Validate data range
- Validate data length

Authentication and Password Management Guidelines (sample)

- Require authentication for all pages and resources, except those specifically intended to be public
- All authentication controls must be enforced on a trusted system (e.g., The server)
- Establish and utilize standard, tested, authentication services whenever possible
- Use a centralized implementation for all authentication controls, including libraries that call external authentication services
- Segregate authentication logic from the resource being requested and use redirection to and from the centralized authentication control
- All authentication controls should fail securely
- All administrative and account management functions must be at least as secure as the primary authentication mechanism
- If your application manages a credential store, it should ensure that only cryptographically strong one-way salted hashes of passwords are stored and that the table/file that stores the passwords and keys is write-able only by the application. (Do not use the MD5 algorithm if it can be avoided)
- Password hashing must be implemented on a trusted system (e.g., The server).
- Validate the authentication data only on completion of all data input, especially for sequential authentication implementations
- Authentication failure responses should not indicate which part of the authentication data was incorrect. For example, instead of “Invalid username” or “Invalid password”, just use “Invalid username and/or password” for both. Error responses must be truly identical in both display and source code

Trust, but Verify





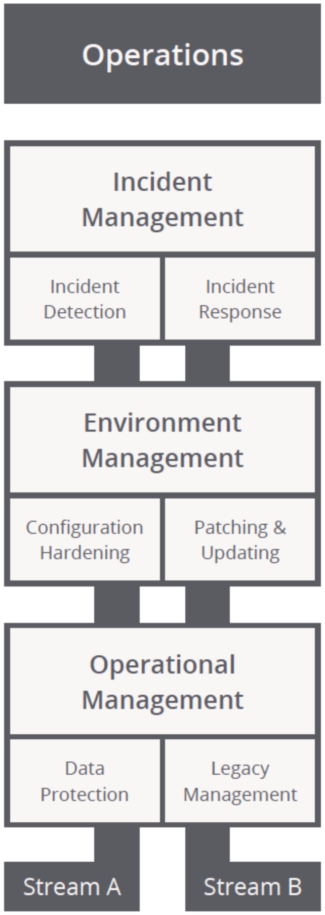
Industry-Standards Bodies Key Management Controls

- KEK must not be less than the DEK (e.g., by encrypting a 256-bit AES key with a 128-bit AES key).
- Any public keys are used by the system in an inventory and the authenticity of all public keys is maintained.
- Where public or white-box keys are not unique per software instantiation, methods and procedures to revoke and replace such keys (or key pairs) exist.
- Any external files or other data elements relied upon for key material (e.g., public TLS certificates), must have clear and sufficient guidance on secure installation provided by the vendor.
- Public keys manually loaded or used as root keys are installed and stored requiring dual control preventing a single user from replacing a key.
- Secret and/or private keys are managed in a way that ensures split knowledge.
- Methods are implemented to "roll" any keys at the end of their defined crypto-period.

Physical Security Architecture (Builder's View)

Focus on the data.	Security Baselines/Hardening Guides
Physical data management includes:	• Layer-3 (firewalls, routers, switches)
• File structures, including record and field structures	• Servers (web, application, database, DC, NTP, etc.)
• File management tools, including directory management	• Operating systems
• Database structures	• Database systems
• DBMSs	Perform periodic firewall (layer-3) rules review - monthly, quarterly, annual, major infrastructure change
Encryption/cryptographic and key management procedures	Perform user certification reviews - monthly
Database Security	Perform privileged user certification reviews – monthly, quarterly
Perform Threat Analysis	Implement SIEM, FIM, AV, WAF alerting with correlation rules and monitoring

SAMM - Operations



	Stream A **Incident Detection**	Stream B **Incident Response**
Maturity level 1 ●○○ Best-effort incident detection and handling	Use available log data to perform best-effort detection of possible security incidents.	Identify roles and responsibilities for incident response.
Maturity level 2 ●●○ Formal incident management process in place	Follow an established, well-documented process for incident detection, with emphasis on automated log evaluation.	Establish a formal incident response process and ensure staff are properly trained in performing their roles.
Maturity level 3 ●●● Mature incident management	Use a proactively managed process for detection of incidents.	Employ a dedicated, well-trained incident response team.

iSecurePrivacy LLC

OWASP

SAMM - Operations

Operations – this involves activities necessary to ensure confidentiality, integrity, and availability (CIA) are maintained throughout the operational lifetime of an application and its associated data.

- **Incident Management** - This practice addresses activities carried out improve the organization’s detection of, and response to, security incidents.
- **Environment Management** - This practice describes proactive activities carried out to improve and maintain the security of the environments in which the organization's applications operate.
- **Operational Management** - This practice focuses on operational support activities required to maintain security throughout the product lifecycle.

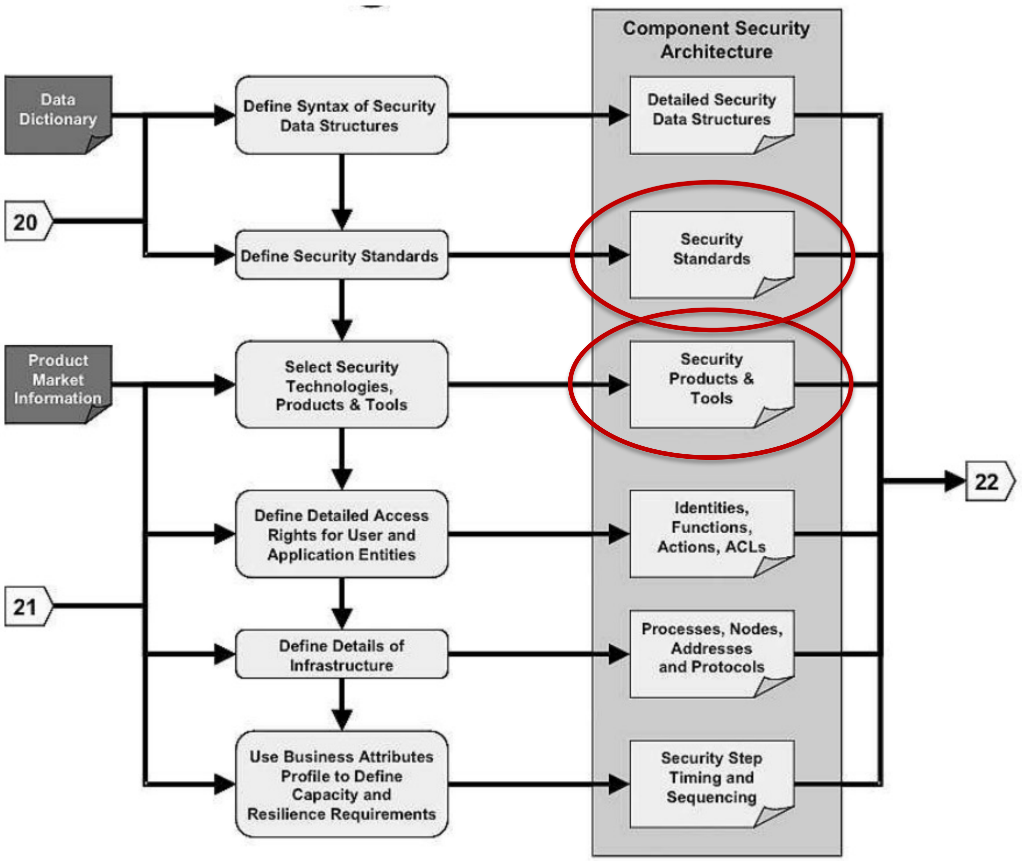
Component Security Architecture (Engineer’s View)

Focus on the tools.

Decide on Security Standards

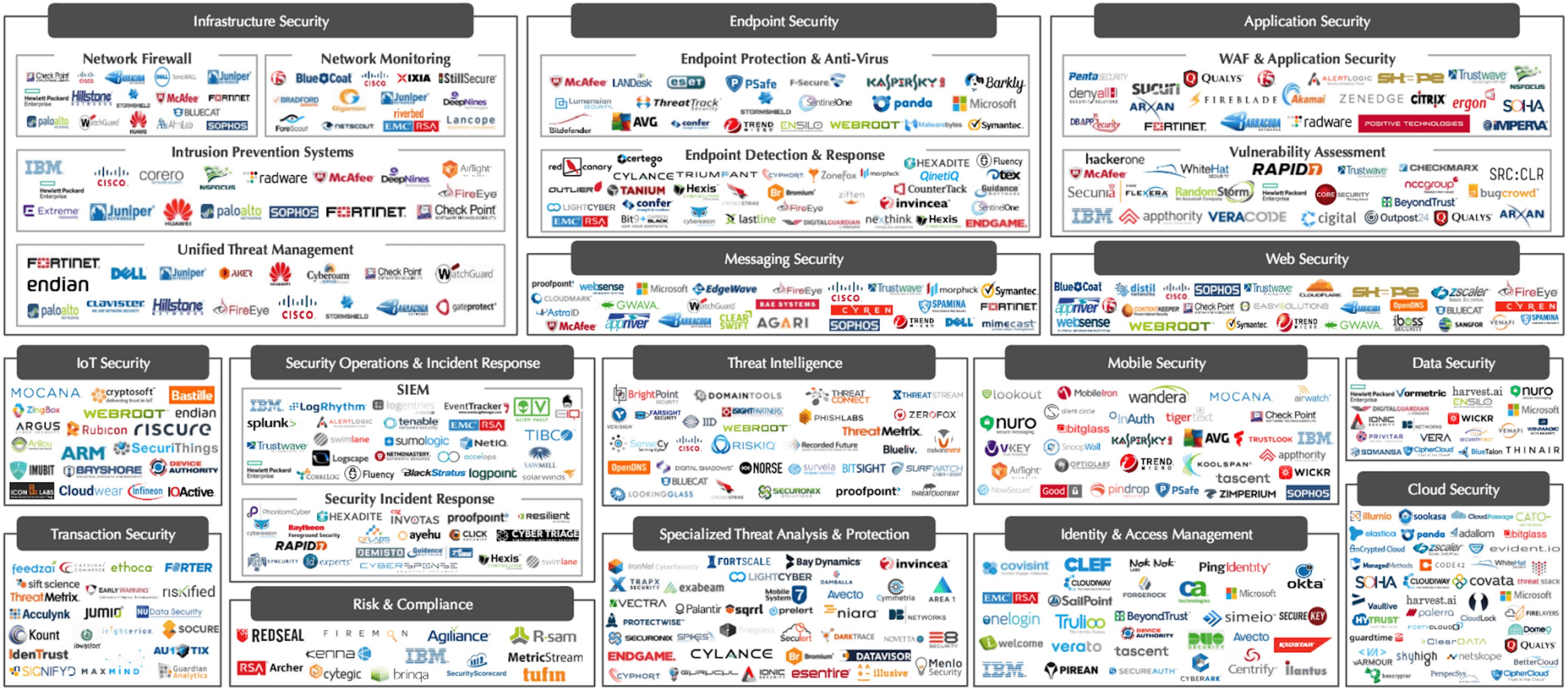
- ISO
- ISACA
- CIS
- NIST
- OWASP
- PCI SSF

Decide on Security Products and Tools


Data Dictionary
20
Product Market Information
21
Define Syntax of Security Data Structures
Define Security Standards
Select Security Technologies, Products & Tools
Define Detailed Access Rights for User and Application Entities
Define Details of Infrastructure
Use Business Attributes Profile to Define Capacity and Resilience Requirements
Component Security Architecture
Detailed Security Data Structures
Security Standards
Security Products & Tools
Identities, Functions, Actions, ACLs
Processes, Nodes, Addresses and Protocols
Security Step Timing and Sequencing
22

Source: Sherwood, Nicholas A. Enterprise Security Architecture: A Business-Driven Approach (p. 2). CRC Press. Kindle Edition.

The Security Sector Is Dynamic And Vast. We Are Ceaseless & Vigilant In Our Coverage.



Infrastructure Security

Network Firewall

Network Monitoring

Intrusion Prevention Systems

Unified Threat Management

Endpoint Security

Endpoint Protection & Anti-Virus

Endpoint Detection & Response

Application Security

WAF & Application Security

Vulnerability Assessment

Messaging Security

Web Security

IoT Security

Security Operations & Incident Response

SIEM

Security Incident Response

Threat Intelligence

Mobile Security

Data Security

Cloud Security

Transaction Security

Risk & Compliance

Specialized Threat Analysis & Protection

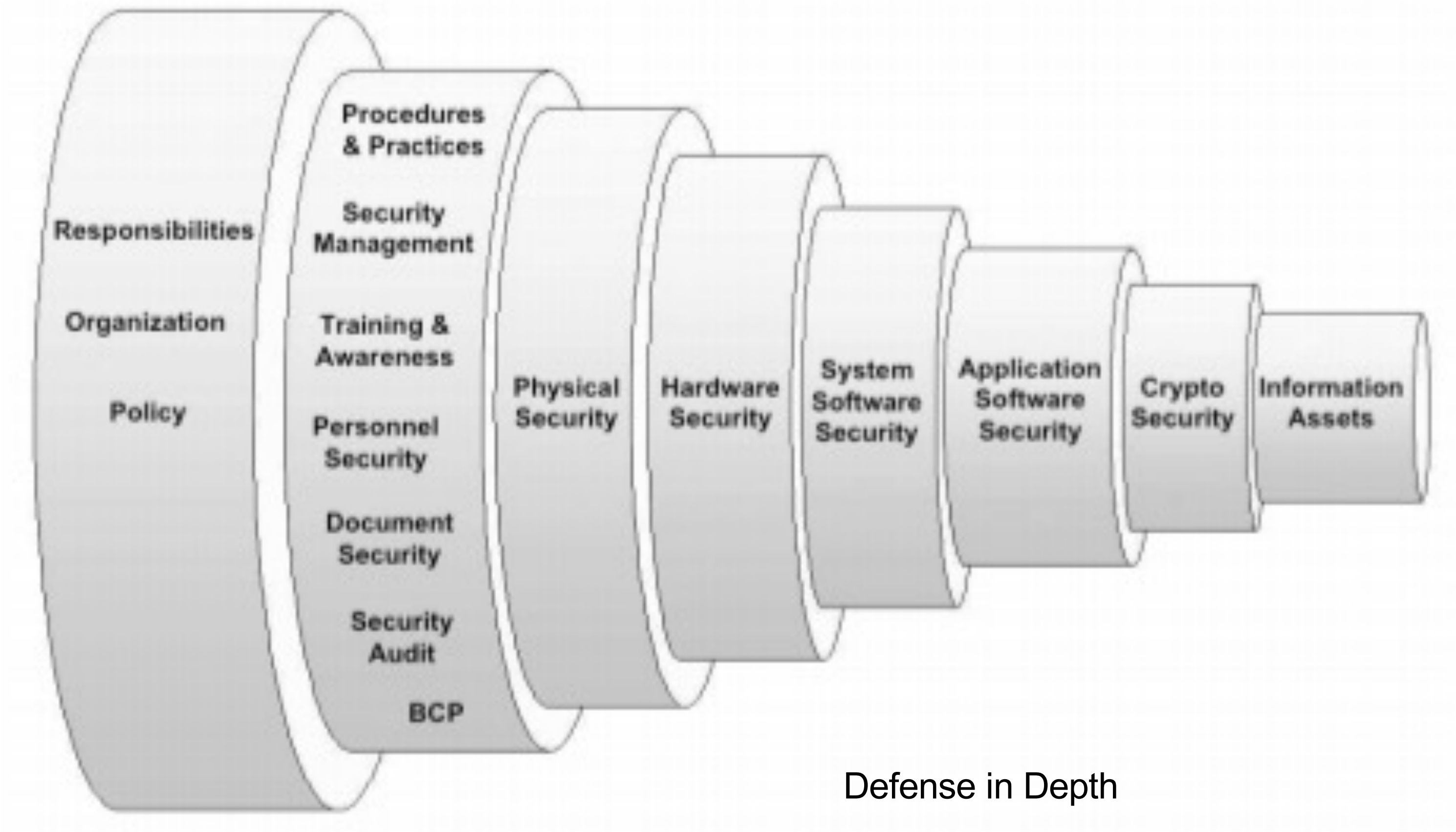
Identity & Access Management

Source: Momentum Partners.

iSecurePrivacy LLC

OWASP

Multi-Layered Security Model



Defense in Depth

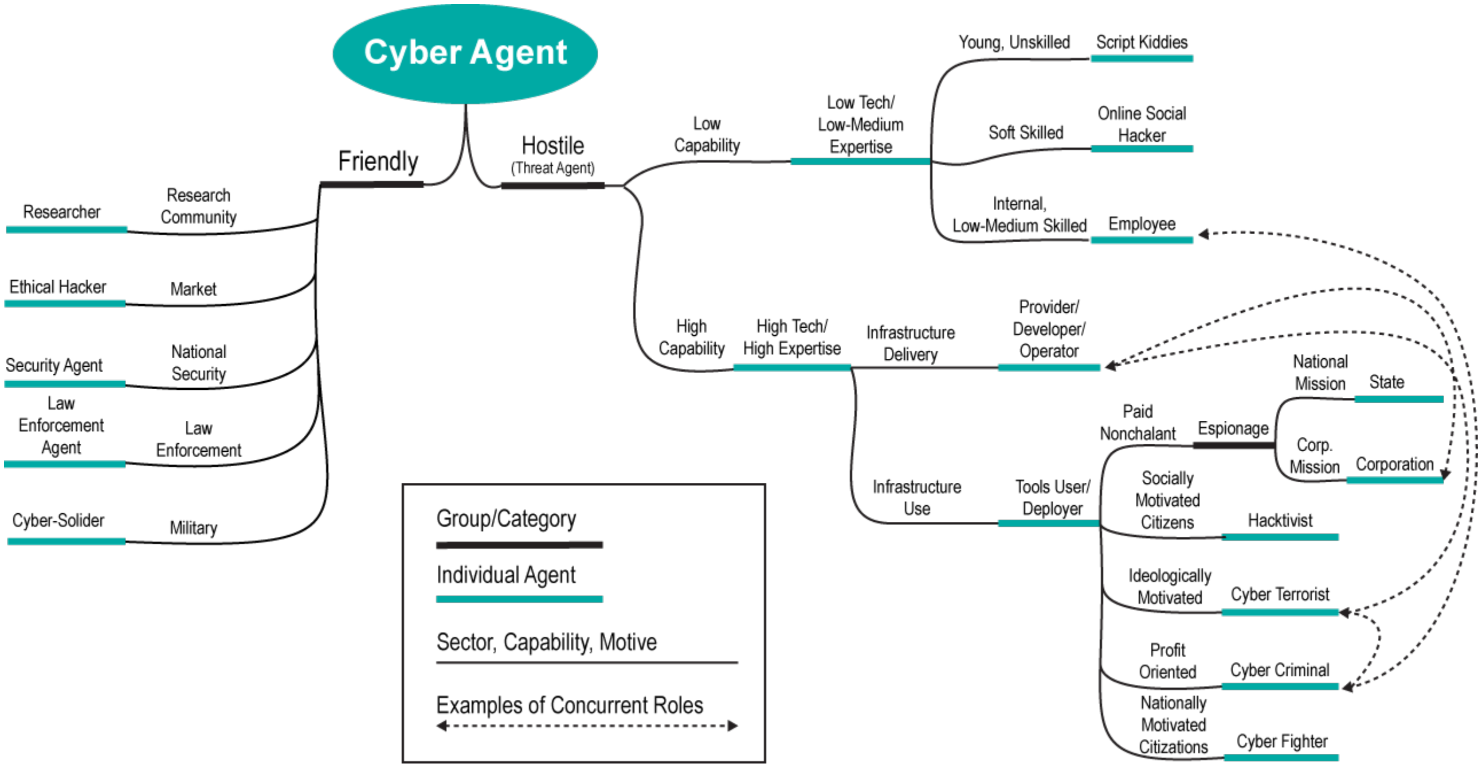
Interdependencies

Modern IT architectures are usually decentralized and deperimeterized, increasing security risk across several fronts, including:

- Cloud-based platforms and services
- Smart and mobile devices
- Third-party products and services
- Weak and unsecured parts of the IT architecture

This interdependent environment means control has been reduced—a change with important impacts on security architecture.

Common Threat Agents



Source: Marinos, Louis, A. Belmonte, E. Rekleitis, "ENISA Threat Landscape 2015," ENISA, January 2016, Greece

Generative AI

- **Google's Bard -** ChatGPT's main competitor is Bard, Google's AI generative AI chatbot. Now Google plans to add Bard into search. In comparison to ChatGPT
- **Baidu's Ernie -** The Chinese search engine Baidu plans to add a chatbot called Ernie. Baidu announced the upcoming change on March 16, 2023, at which point the initial showing disappointed investors.
- **DeepMind's Sparrow -** DeepMind focuses more on research and has not yet come out with a public-facing chatbot. DeepMind does have Sparrow, a chatbot designed specifically to help AI communicate in a way that is "helpful, correct and harmless." DeepMind founder Demis Hassabis told The Independent in January 2023 that DeepMind may release a private beta version of Sparrow later in 2023.
- **Meta's BlenderBot -** Meta released BlenderBot in August 2022. The prototype BlenderBot from the company behind Facebook focuses on being able to chat, providing short, conversational replies rather than full paragraphs.
- **What about Apple? -** According to The New York Times, Apple is working on leveraging the tech it has, especially Siri, to create a ChatGPT rival. More information about what the final product might look like is thin on the ground for now.

OpenSource Risk

Fake Researcher Profiles Spread Malware through GitHub Repositories as PoC Exploits
June 14, 2023

https://thehackernews.com/2023/06/fake-researcher-profiles-spread-malware.html

At least half of dozen GitHub accounts from fake researchers associated with a fraudulent cybersecurity company have been observed pushing malicious repositories on the code hosting service.

All seven repositories, which are still available as of writing, claim to be a proof-of-concept (PoC) exploit for purported zero-day flaws in Discord, Google Chrome, and Microsoft Exchange Server.

VulnCheck, which discovered the activity, said, "the individuals creating these repositories have put significant effort into making them look legitimate by creating a network of accounts and Twitter profiles, pretending to be part of a non-existent company called High Sierra Cyber Security."

Security Standards

- International Organization for Standards (ISO) 27000
- ISACA COBIT
- The World Wide Web Consortium (W3C)
- OWASP Top 10
- Common Weakness Enumeration (CWE)
- US Federal Government (NIST)
- Center of Internet Security (CIS)
- International Security Forum (ISF)
- PCI Secure Software Framework (PCI SSF)
- Internal Security Standards (your company)
- Vendor Standards

Why Security Requirements Are Hard

1. Internally developed solutions are superior (pride/narcissism)
2. Jealous that proposed solutions is better than the one internally developed (jealous)
3. A prophet is not accepted in his own country (lack of respect)
4. Fear of looking bad or being embarrassed for internal solution (fear of being embarrassed)
5. They just don’t like you (previous experience with you or inherently who you are – e.g., auditor)
6. What they do is all they know (hammer-vs-nail)
7. Participants suffer from the NIH Syndrome

Not-Invented-Here (NIH) Syndrome

No thanks!
We are too busy

“THE MAN WHO SAYS HE CAN,

AND THE MAN WHO SAYS HE CANNOT

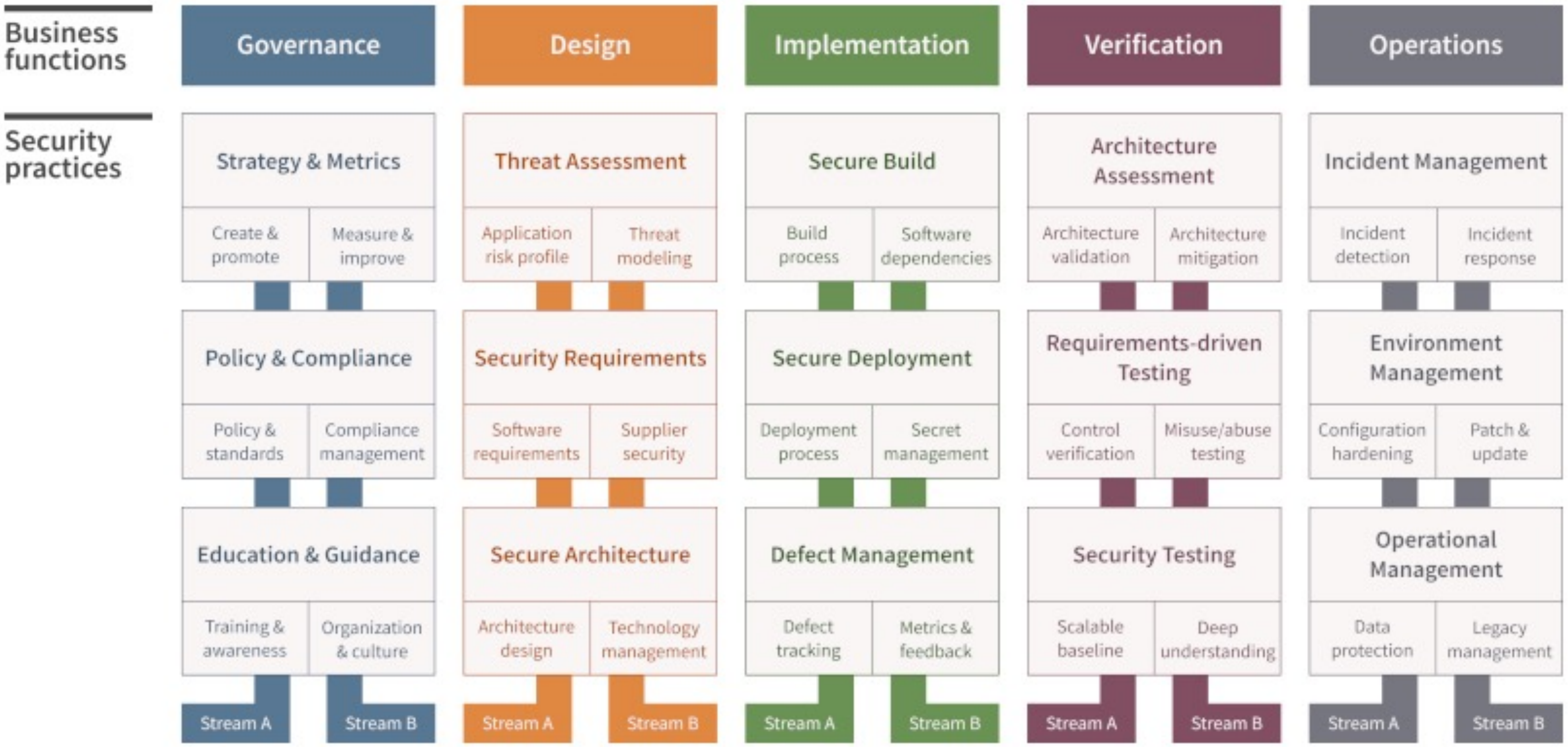
ARE BOTH CORRECT”

-- Confucius

Deploying a cybersecurity architecture

1. **Assessment**: Assess the organization's current security posture to identify potential risks, vulnerabilities, and areas for improvement.
2. **Planning**: Develop a comprehensive cybersecurity plan that aligns with the organization's goals, priorities, and budget.
3. **Design**: Develop a detailed design of the cybersecurity architecture, including the different layers of security controls and how they will work together.
4. **Implementation**: Deploy the cybersecurity architecture, which may involve implementing new security controls and configuring existing ones.
5. **Testing**: Conduct testing to ensure that the cybersecurity architecture is working as intended and that all security controls are properly configured.
6. **Maintenance**: Regularly maintain and update the cybersecurity architecture to ensure that it remains effective against emerging threats and that it aligns with the organization's changing needs and priorities.
7. **Deploying** a cybersecurity architecture requires a coordinated effort between IT, security teams, and business stakeholders.

OWASP SAMM v2.0



Business functions	Governance		Design		Implementation		Verification		Operations	
Security practices	Strategy & Metrics		Threat Assessment		Secure Build		Architecture Assessment		Incident Management	
	Create & promote	Measure & improve	Application risk profile	Threat modeling	Build process	Software dependencies	Architecture validation	Architecture mitigation	Incident detection	Incident response
	Policy & Compliance		Security Requirements		Secure Deployment		Requirements-driven Testing		Environment Management	
	Policy & standards	Compliance management	Software requirements	Supplier security	Deployment process	Secret management	Control verification	Misuse/abuse testing	Configuration hardening	Patch & update
	Education & Guidance		Secure Architecture		Defect Management		Security Testing		Operational Management	
	Training & awareness	Organization & culture	Architecture design	Technology management	Defect tracking	Metrics & feedback	Scalable baseline	Deep understanding	Data protection	Legacy management
	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B	Stream A	Stream B

iSecurePrivacy LLC

The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization.

REFERENCES

NIST SP 800-160

- https://csrc.nist.gov/publications/detail/sp/800-160/vol-1-rev-1/final
- https://csrc.nist.gov/publications/detail/sp/800-160/vol-2-rev-1/final

AWS Security by Design Framework

- https://aws.amazon.com/compliance/security-by-design/

SABSA Framework

- https://sabsa.org/sabsa-executive-summary/

COBIT

- https://www.isaca.org/resources/cobit

PCI SSF

- https://www.pcisecuritystandards.org/document_library/

OWASP SAAM

- https://owasp.org/2020/02/11/SAMM-v2

THANK YOU!

BIO

Miguel (Mike) O. Villegas is the President and Founder of iSecurePrivacy, LLC. His is currently CISO for TRISTAR Insurance Group, the largest privately-owned Third-Party Administrator (TPA) in USA. He is also current CTO for Xahive, a cybersecurity software firm specializing in security awareness training and compliance software. He was previously SVP for K3DES, LLC., Director of Information Security at Newegg, Inc., and Contributing Writer for SearchSecurity.com -TechTarget with over 150 articles.

He has over 35 years of Information Systems Security and IT audit experience. Mike was previously Vice President & Technology Risk Manager for Wells Fargo Services responsible for IT Regulatory Compliance and was also a partner at Arthur Andersen and Ernst & Young for their information systems security and IS audit groups over a span of nine years.

Mike was president of the LA ISACA Chapter during 2010-2012 and president of the SF ISACA Chapter during 2005-2006. He was the SF Fall Conference Co-Chair from 2002–2007 and also served two years as Vice President on the Board of Directors for ISACA International. He is a CISA, CISSP, CDPSE, CEH, CSX|F, CSX|A, and ISO/IEC Lead Implementer. As of February 1, 2023, he has fifteen (15) years experience as a PCI QSA, PA-QSA and three (3) years experience as a PCI SSF Assessor. Mike is currently Certification Chair for the ISACA LA Chapter and has taught CISA review courses for over 25 years.