SECURITY

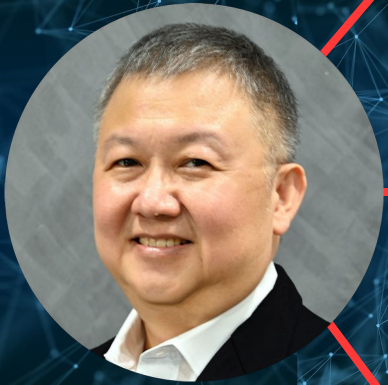
SHIFT ◀◀◀ LEFTMOST

<DEV>
MOUNTAIN
TECH FESTIVAL

Secure Architecture

Narudom Roongsiriwong, CISSP, CCSK
March 20, 2022

Narudom Roongsiriwong



SVP, Head of Digital Architecture

krungsri
กรุงศรี
A member of MUFG
a global financial group

APAC Research Advisory Council Member

Co-Chair, Hybrid Cloud Security Working Group

CSA
cloud security alliance℠

Consultant, Thailand Chapter

OWASP
Open Web Application Security Project

What Is “Shift Left” on Security?

> *“By better integrating information security (InfoSec) objectives into daily work, teams can achieve higher levels of software delivery performance and build more secure systems. This idea is also known as **shifting left**, because concerns, including security concerns, are addressed earlier in the software development lifecycle (that is, left in a left-to-right schedule diagram).”*
>
> *Cloud Architecture Center → DevOps → Guides*
> *DevOps tech: Shifting left on security*

https://cloud.google.com/architecture/devops/devops-tech-shifting-left-on-security

Jan 4, 2021, 10:30am EST | 2,242 views

A Modern Shift-Left Security Approach

Richard Seiersen Forbes Councils Member
Forbes Technology Council COUNCIL POST | Membership (Fee-Based)
Innovation

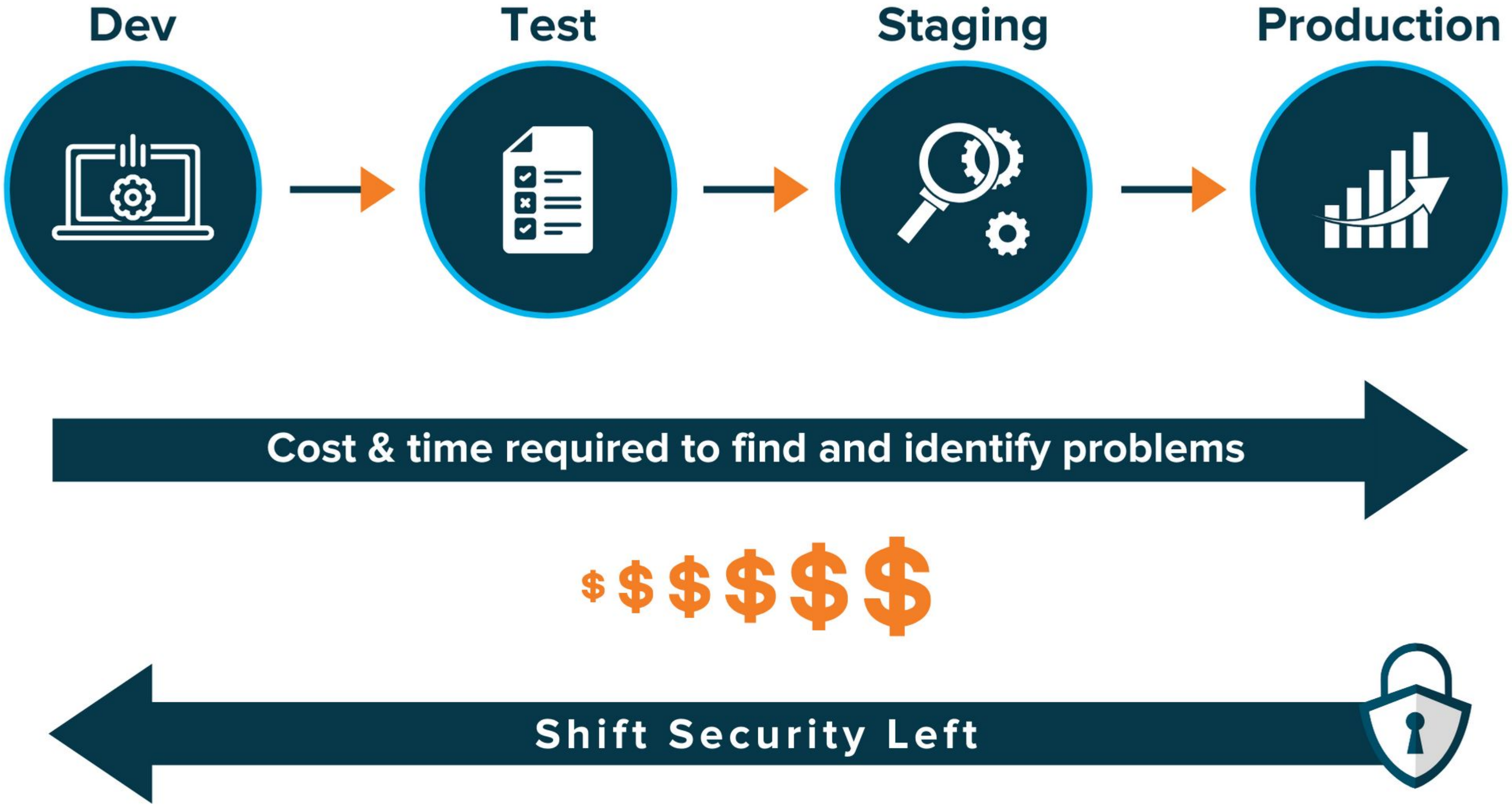
CEO and Co-Founder of Soluble, former serial CISO, and author of "How To Measure Anything in Cybersecurity Risk"

GETTY

The concept of shifting security left is not new, but historically this has meant little more than inserting security processes in the middle of development and slowing everything down. In this article, I'll describe older shift-left methods that have not worked — and how a modern approach to shifting left can have a high impact on risk reduction and create a healthy balance of freedom and responsibility for cloud-native development teams.

I believe CISOs have no choice but to embrace this industry change. They must enable development teams to move faster while simultaneously reducing the

DevSecOps’ Security “Shift Left”



Dev

Test

Staging

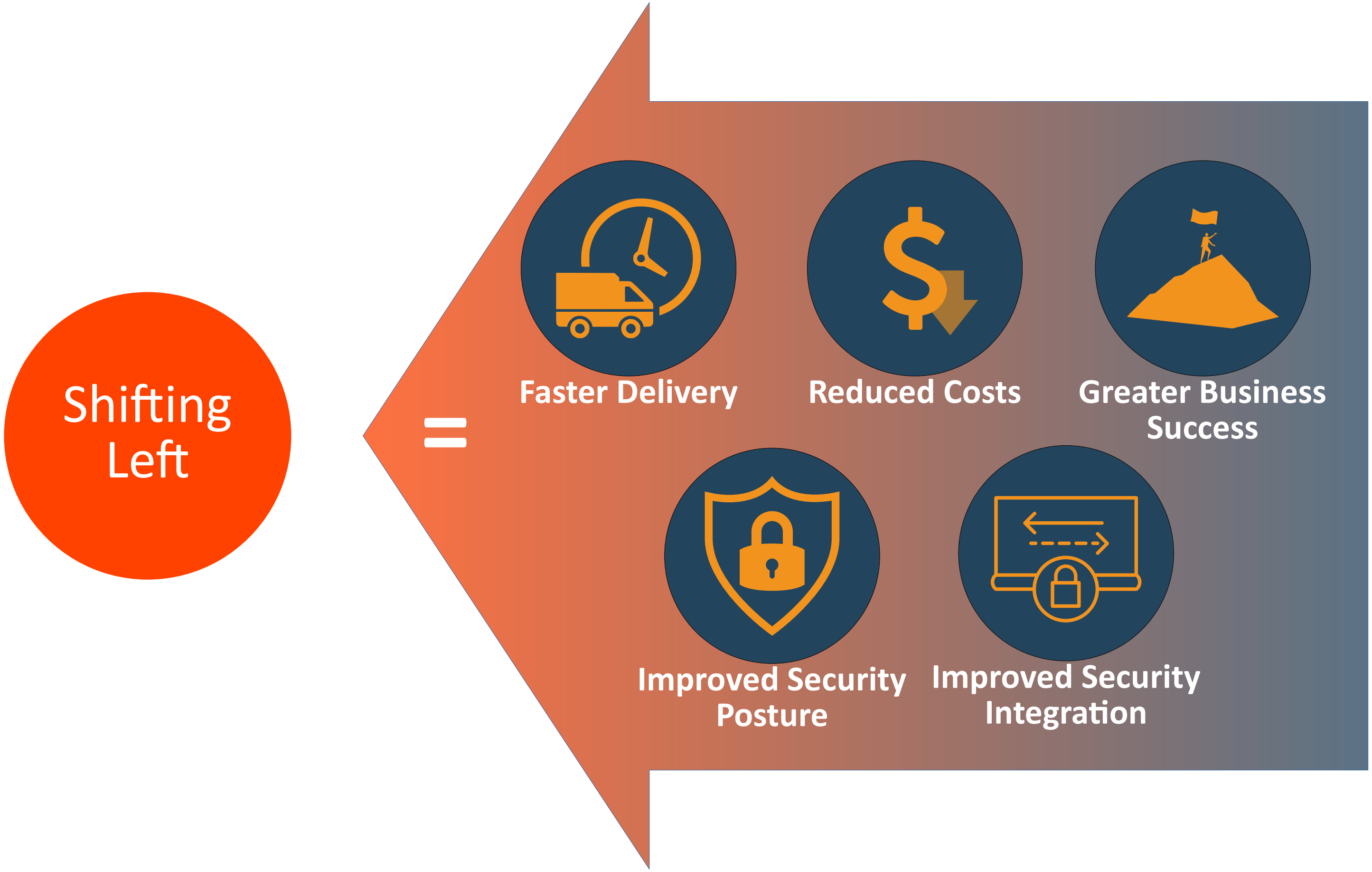
Production

Cost & time required to find and identify problems

$ $ $ $ $ $

Shift Security Left

Benefits of Shifting Left



Shifting Left

=

Faster Delivery

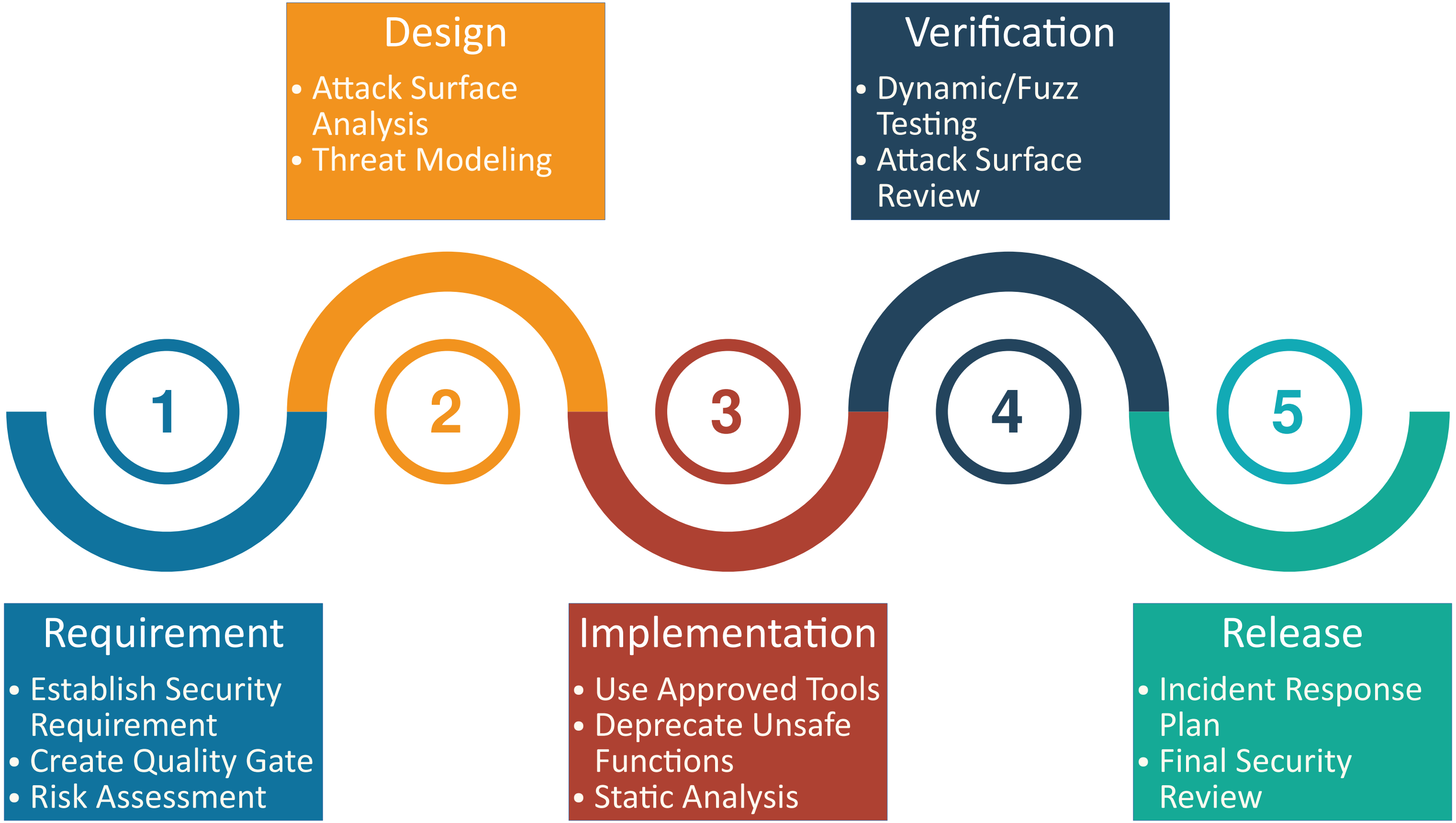
Reduced Costs

Greater Business Success

Improved Security Posture

Improved Security Integration

Security Activities in Software Development



Requirement

- Establish Security Requirement
- Create Quality Gate
- Risk Assessment

Design

- Attack Surface Analysis
- Threat Modeling

Implementation

- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

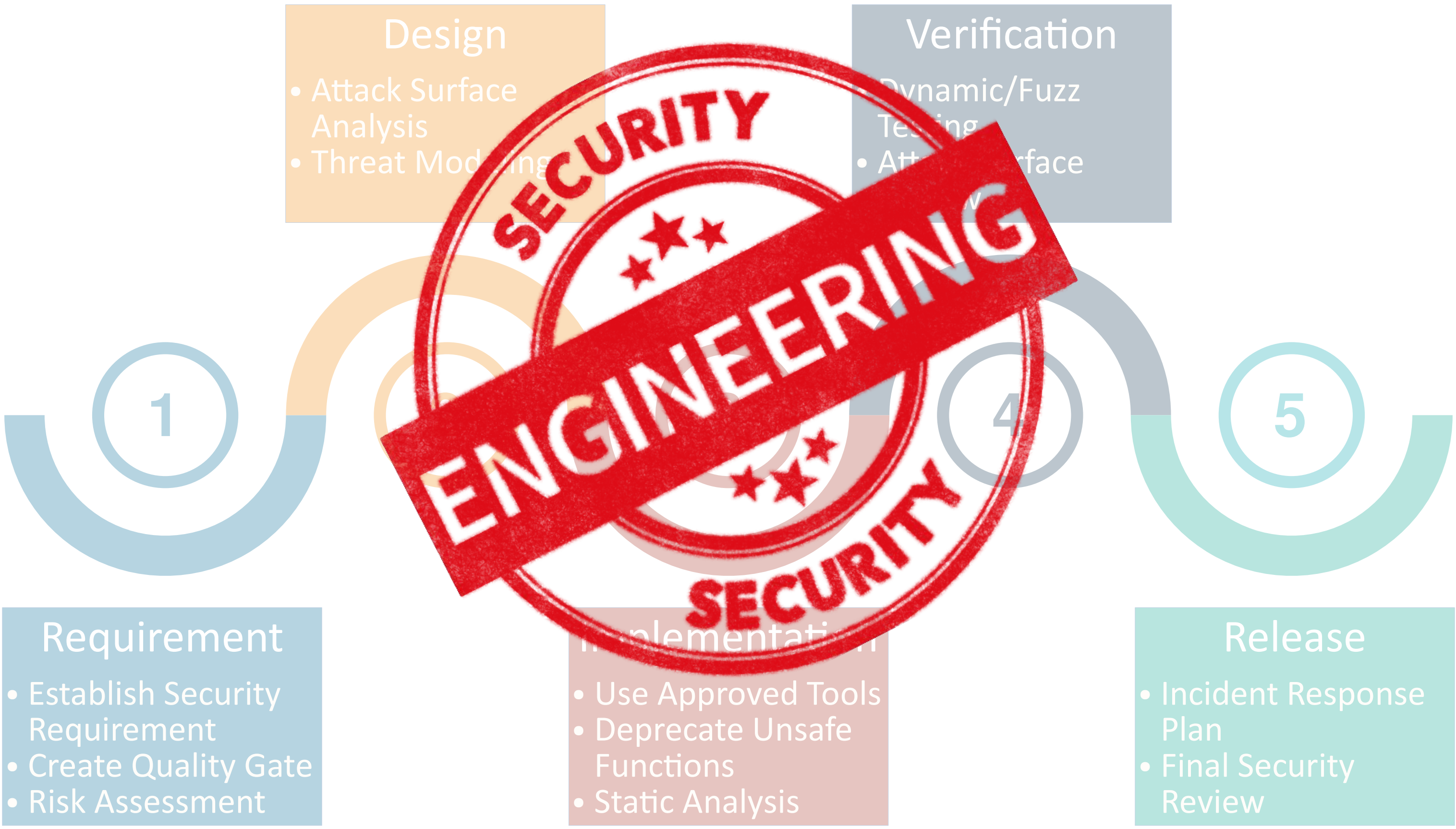
Verification

- Dynamic/Fuzz Testing
- Attack Surface Review

Release

- Incident Response Plan
- Final Security Review

Security Activities in Software Development



SECURITY ENGINEERING SECURITY

Requirement

- Establish Security Requirement
- Create Quality Gate
- Risk Assessment

Design

- Attack Surface Analysis
- Threat Modeling

Implementation

- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

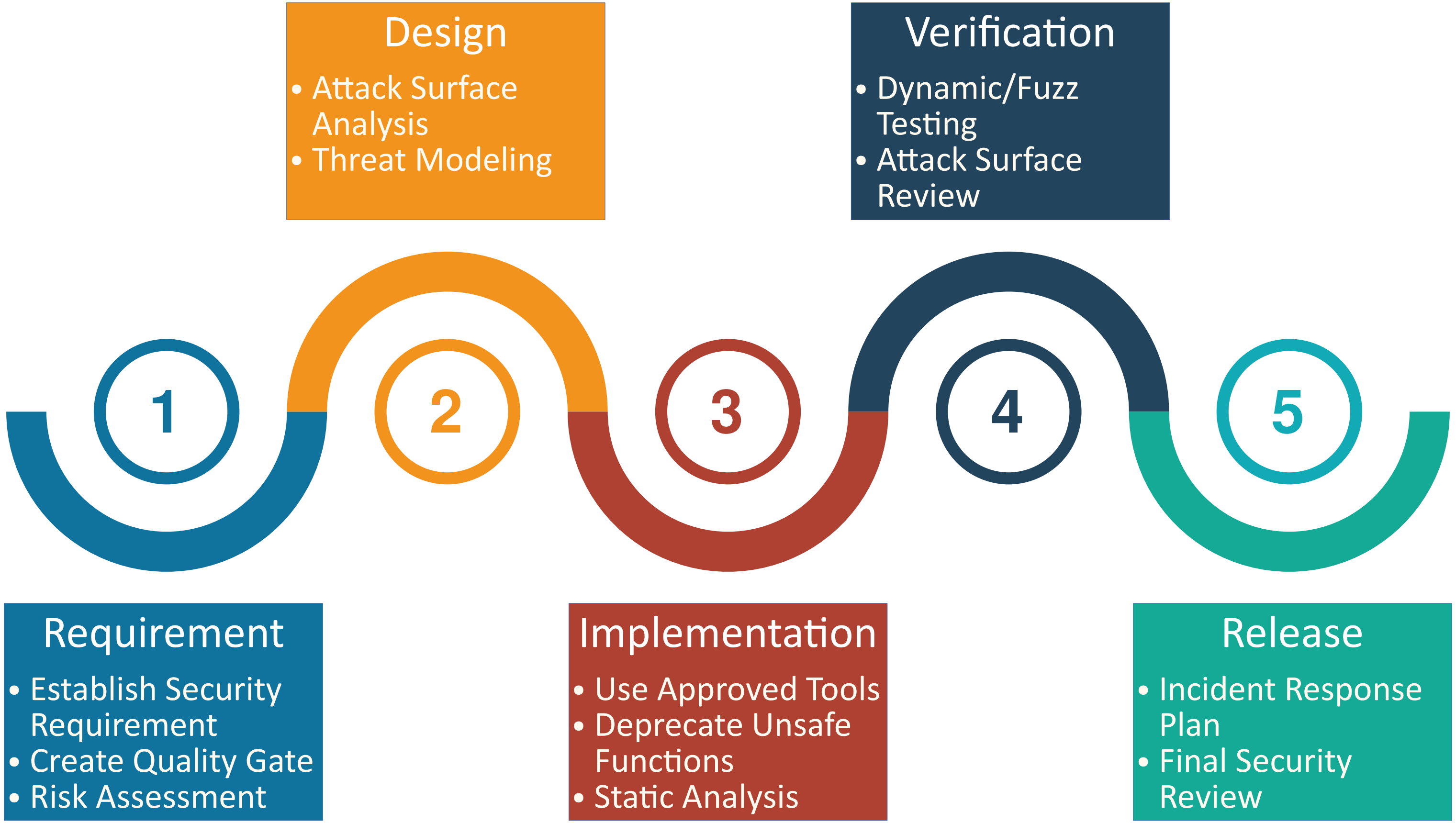
Verification

- Dynamic/Fuzz Testing
- Attack Surface Review

Release

- Incident Response Plan
- Final Security Review

What is the Leftmost?



Requirement

- Establish Security Requirement
- Create Quality Gate
- Risk Assessment

Design

- Attack Surface Analysis
- Threat Modeling

Implementation

- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

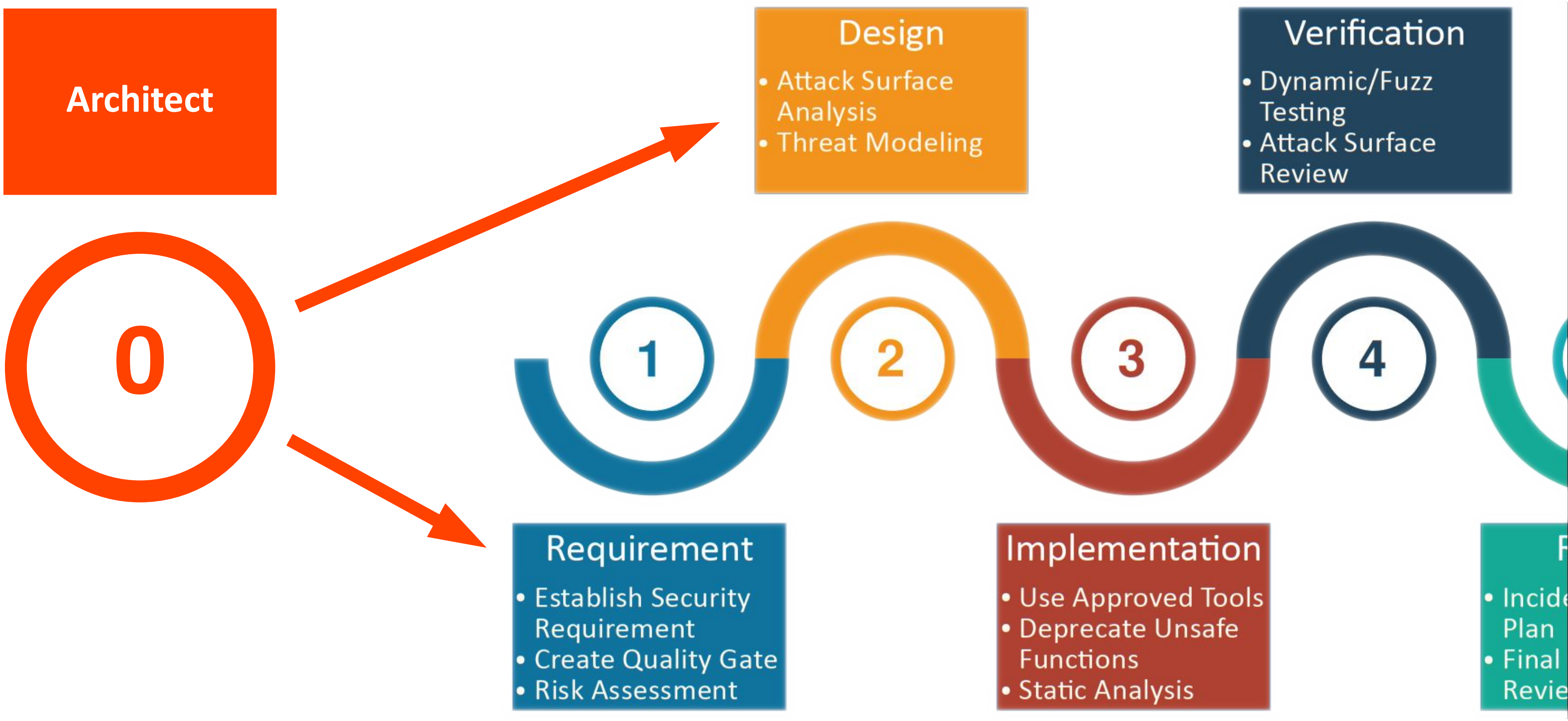
Verification

- Dynamic/Fuzz Testing
- Attack Surface Review

Release

- Incident Response Plan
- Final Security Review

What is the Leftmost?



Architect

0

Design
- Attack Surface Analysis
- Threat Modeling

Verification
- Dynamic/Fuzz Testing
- Attack Surface Review

Requirement
- Establish Security Requirement
- Create Quality Gate
- Risk Assessment

Implementation
- Use Approved Tools
- Deprecate Unsafe Functions
- Static Analysis

F
- Incide
  Plan
- Final
  Revie

The Enterprise

Architecting: Enterprise Application Infrastructure

- Identity and access management
  - Internal users
  - Customers
- Cryptographic and key management
- Data sharing platform
- Security requirement
- Security quality gate guideline
- Security patterns (design & integration patterns)
- Threat and vulnerability catalog

Identity and Access Management

IAM & CIAM

	MANAGE IDENTITIES FOR:	USER EXPERIENCE	IDENTITY PROVIDERS	CAPABILITIES
IAM	Internal users: • Employees	Needs to meet a minimum standard – employees will have training.	• Enterprise Directories	SSO, directories, SaaS app access control, authentication, MFA, employee provisioning, delegated admin.
CIAM	External users: • Customers • Contractors • Citizens • Partners • Things • APIs	Customers expect a good UX and will not accept services that require training.	• Social Login • Enterprise Directories • Bank IDs • Regional eIDs • Digital ID Schemes	Registration, Customer SSO, identity proofing, strong authentication & MFA, authorisation workflows, consent management, delegated authority, identity data and attribute directories.

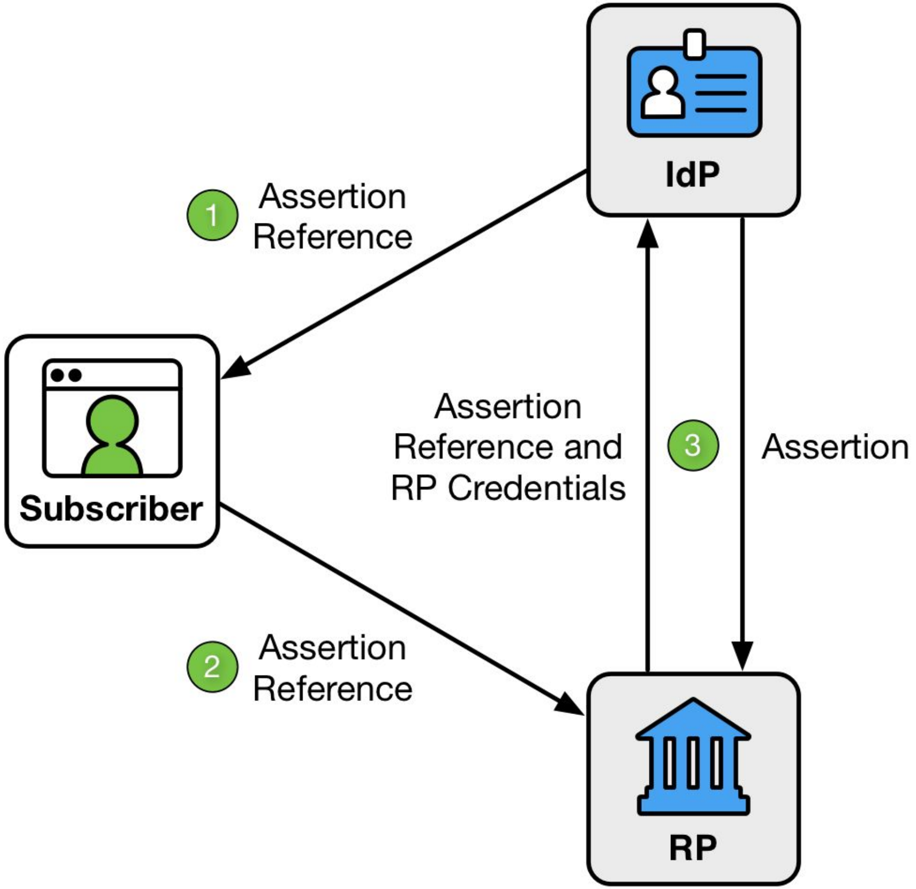
Identity and Access Management

IAM & CIAM

	SCALABILITY	IDENTITIES MANAGED BY:	PRIVACY, SECURITY & DATA	REVENUE & COST SAVINGS
IAM	Predictable number of users, typically significantly fewer employees than customers	HR/IT – can be relatively manual.	The organisation owns and operates the data of a user.	Not designed to add revenue. Implemented to achieve security, operational efficiency and internal compliance goals.
CIAM	Significantly more users than IAM, must be able to scale at a higher rate	User themselves (self-service), plus support desks. Must support complex delegation workflows.	The external user must be in control of their own data. Must help achieve GDPR. Consider Identity data a high target risk for breach attacks.	Will positively impact customer conversion and engagement goals. Will help reduce support desk costs and deliver customer-facing operational efficiency.

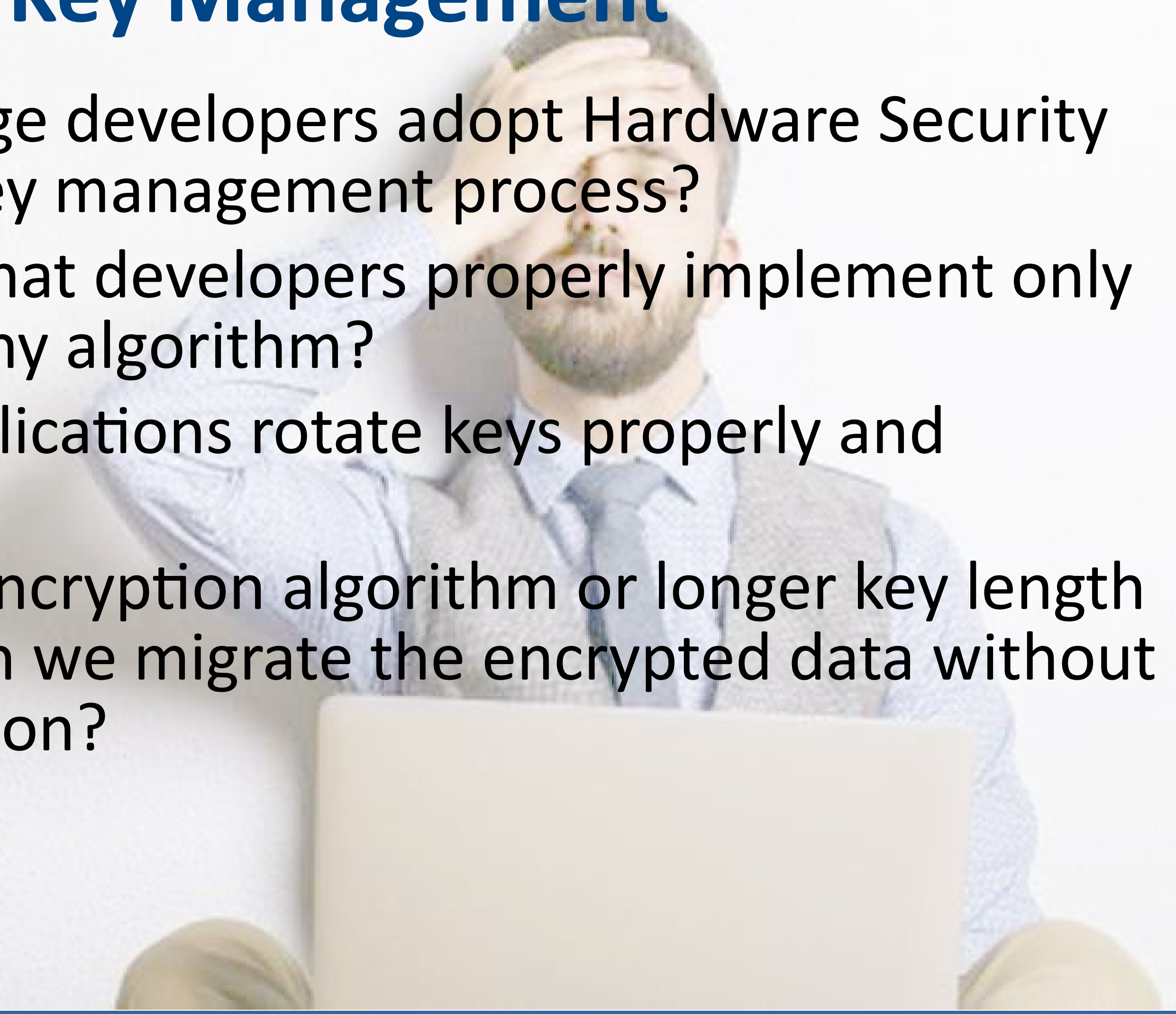
Identity and Access Management

Federated Identity Management: Back Channel Assertion



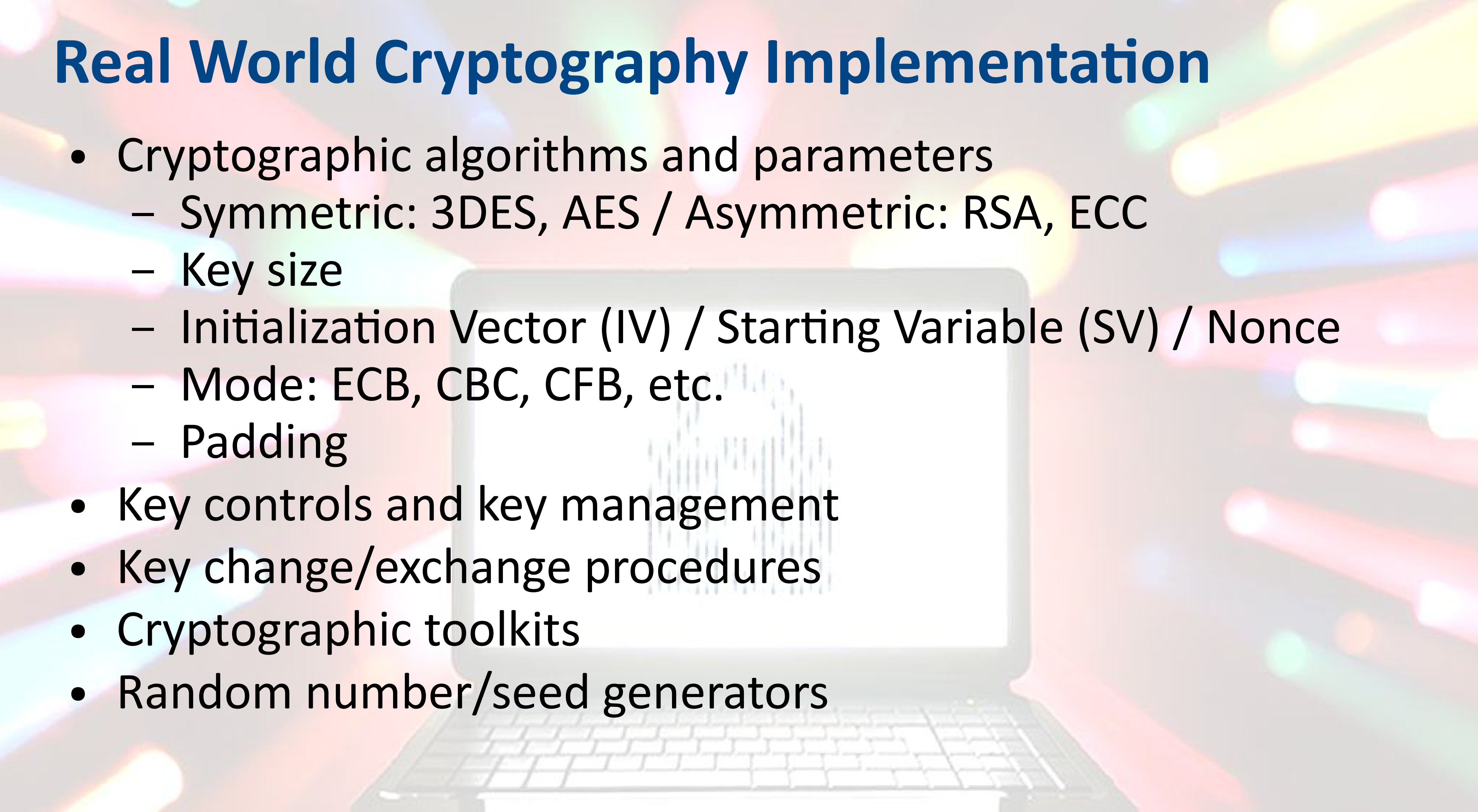
- The subscriber is given an assertion reference to present to the RP, generally through the front channel.
- The assertion reference itself contains no information about the subscriber and SHALL be resistant to tampering and fabrication by an attacker.

Cryptographic and Key Management



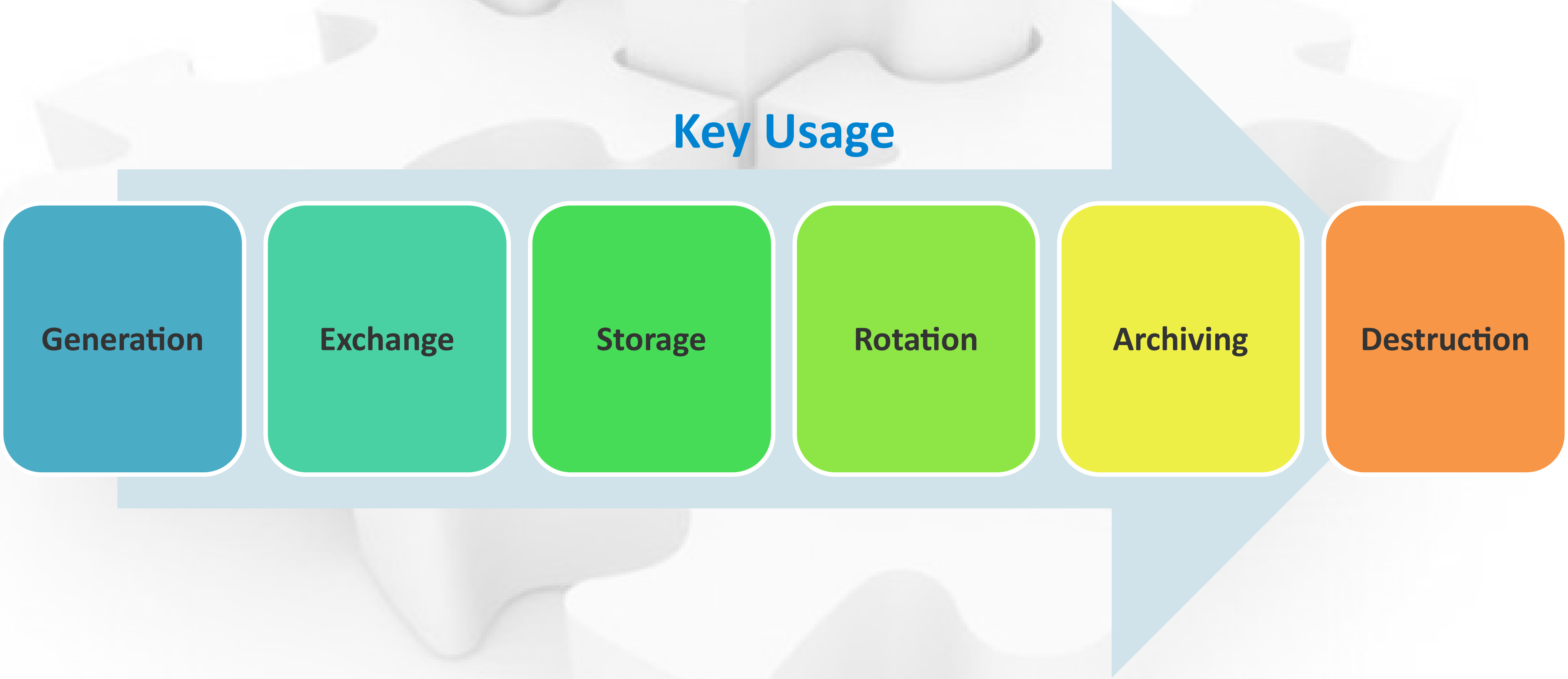
- How can we encourage developers adopt Hardware Security Module (HSM) and key management process?
- How can we ensure that developers properly implement only approved cryptography algorithm?
- How can we help applications rotate keys properly and correctly?
- If we need stronger encryption algorithm or longer key length in the future, how can we migrate the encrypted data without application modification?

Real World Cryptography Implementation



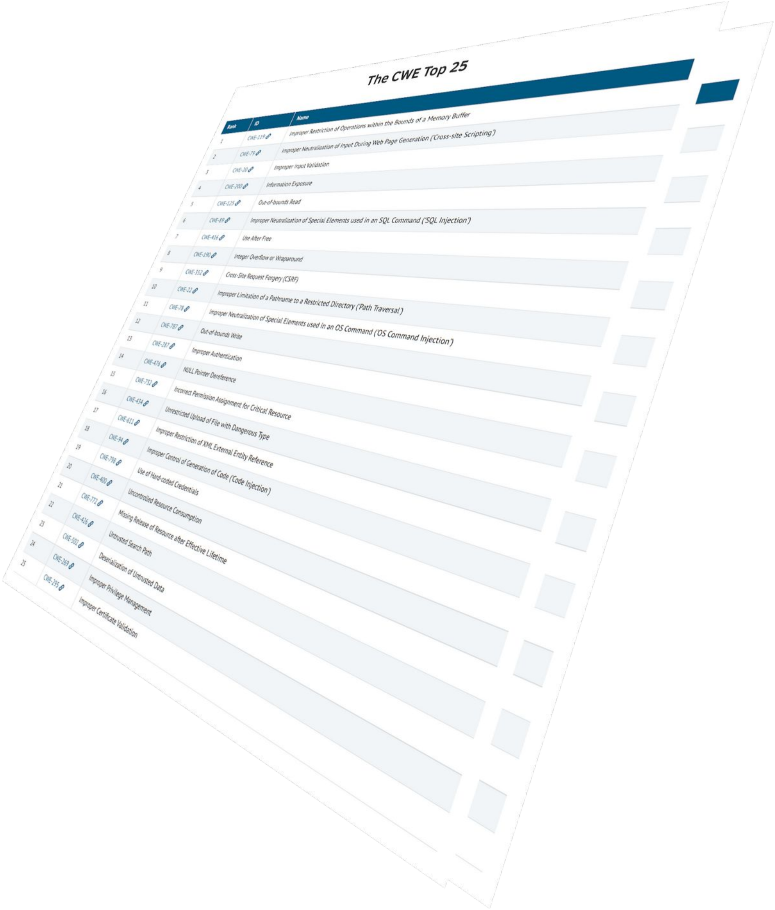
- Cryptographic algorithms and parameters
  - Symmetric: 3DES, AES / Asymmetric: RSA, ECC
  - Key size
  - Initialization Vector (IV) / Starting Variable (SV) / Nonce
  - Mode: ECB, CBC, CFB, etc.
  - Padding
- Key controls and key management
- Key change/exchange procedures
- Cryptographic toolkits
- Random number/seed generators

Key Management Framework



Threat and Vulnerability Catalog

- SANS CWE Top 25
- OWASP Top 10 – 2021
- OWASP Mobile Top 10
- OWASP Application Security Verification Standard (ASVS)



The CWE Top 25

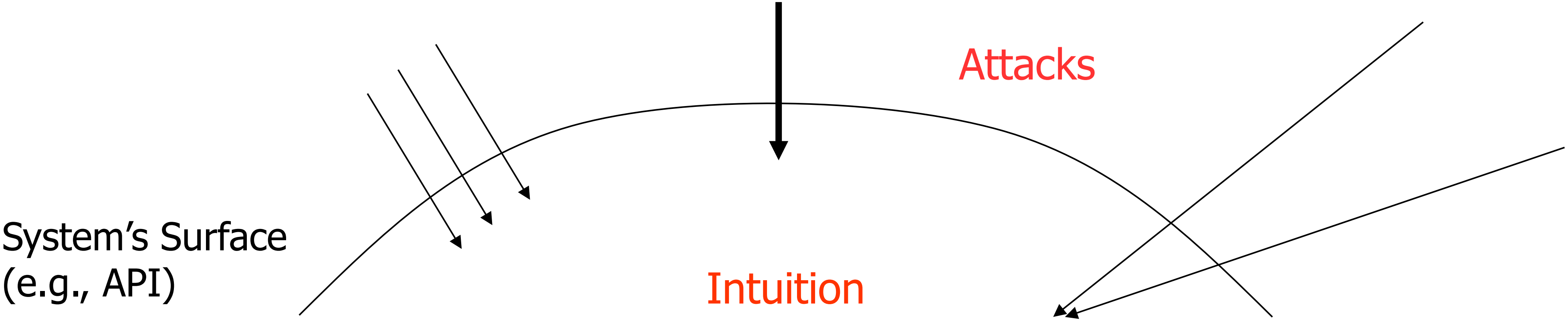
Rank	ID	Name
1	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
3	CWE-20	Improper Input Validation
4	CWE-200	Information Exposure
5	CWE-125	Out-of-bounds Read
6	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
7	CWE-416	Use After Free
8	CWE-190	Integer Overflow or Wraparound
9	CWE-352	Cross-Site Request Forgery (CSRF)
10	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
11	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
12	CWE-787	Out-of-bounds Write
13	CWE-287	Improper Authentication
14	CWE-476	NULL Pointer Dereference
15	CWE-732	Incorrect Permission Assignment for Critical Resource
16	CWE-434	Unrestricted Upload of File with Dangerous Type
17	CWE-611	Improper Restriction of XML External Entity Reference
18	CWE-94	Improper Control of Generation of Code ('Code Injection')
19	CWE-798	Use of Hard-coded Credentials
20	CWE-400	Uncontrolled Resource Consumption
21	CWE-772	Missing Release of Resource after Effective Lifetime
22	CWE-426	Untrusted Search Path
23	CWE-502	Deserialization of Untrusted Data
24	CWE-269	Improper Privilege Management
25	CWE-295	Improper Certificate Validation

Architecting: Project Specific

- Architecture Risk Assessment
  - Architecture security review
  - Attack surface analysis
  - Threat modeling
- Security Architecture and Design
  - Security solutions in response to business and compliance requirements
  - Mitigation and controls in response to risk assessment
- Security Implementation
- Security Operations and Monitoring Integration

Attack Surface

A software or application’s attack surface is the measure of its exposure of being exploited by a threat agent, i.e., weaknesses in its entry and exit points that a malicious attacker can exploit to his or her advantage.



Intuition

Reduce the ways attackers can penetrate surface

⇨ Increase system’s security

Attack Surface Analysis

- Attack Surface Analysis helps you to:
  - Identify what functions and what parts of the system you need to review/test for security vulnerabilities
  - Identify high risk areas of code that require defense-in-depth protection - what parts of the system that you need to defend
  - Identify when you have changed the attack surface and need to do some kind of threat assessment
- Goal attack surface analysis is to reduce the attack surface by:
  - Lower privilege
  - Turn features off
  - Defense in depth

What Is Threat Modeling?

Threat modeling is an approach for analyzing the security system. It is a structured approach that enables you to identify, quantify, and address the security risks associated with a system.

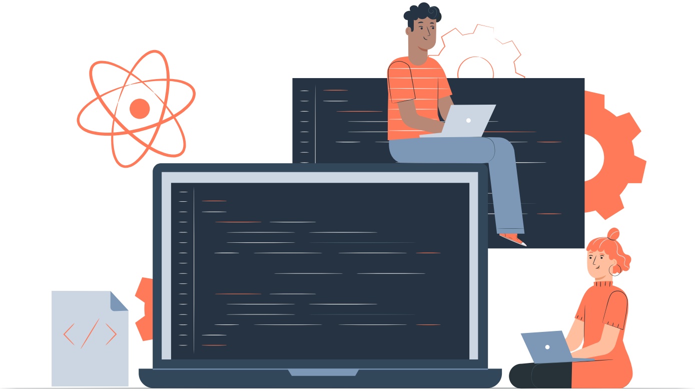
Threat Model Typically Includes

- Description of the subject to be modeled
- Assumptions that can be checked or challenged in the future as the threat landscape changes
- Potential threats to the system
- Actions that can be taken to mitigate each threat
- A way of validating the model and threats, and verification of success of actions taken

Threat Modeling: Four Question Framework

What are we working on?

What can go wrong?



What are we going to do about it?

Did we do a good job?

Q&A