RABET-V Program Description

Version 0.1.0

April 30, 2020

**Reviewing and Editing this Document**

This document was automatically generated from the [RABET-V GitHub Repository](https://github.com/it-dept-cis/RABET-V-Pilot) without significant editing. Many of the links in this document were originally links to other GitHub pages within the repository. As such, these links will not resolve and that is ok.

Also, when you do make changes, please track your changes. We will be incorporating them back into the GitHub repository as our official source of RABET-V information.

Contents

[1 The RABET-V Program 3](#_Toc39478696)

[1.1 Steering Committee 3](#_Toc39478697)

[1.2 Pilot Participants 4](#_Toc39478698)

[1.3 Technology Advisory Committee 4](#_Toc39478699)

[1.4 Program Administration and Research 5](#_Toc39478700)

[1.5 Pilot Process 5](#_Toc39478701)

[1.6 Pilot Research Questions 5](#_Toc39478702)

[2 RABET-V Program 6](#_Toc39478703)

[2.1 Introduction 7](#_Toc39478792)

[2.2 RABET-V Administrator 7](#_Toc39478794)

[2.3 Registered Technology Providers 8](#_Toc39478795)

[2.4 Subscribers 9](#_Toc39478796)

[2.5 RABET-V Public Portal 9](#_Toc39478798)

[3 RABET-V Security Services 10](#_Toc39478820)

[4 RABET-V Maturity Indexes 11](#_Toc39478821)

[5 RABET-V Activities 11](#_Toc39478822)

[5.1 Provider Submission 14](#_Toc39478826)

[5.2 Submission Review Process 17](#_Toc39478827)

[5.3 Process Assessment 19](#_Toc39478828)

[5.4 Architecture Review Methodology 22](#_Toc39478829)

[5.5 Security Claims Validation 25](#_Toc39478830)

[5.6 Testing Rules Determination 30](#_Toc39478831)

[5.7 Product Verification Activity 33](#_Toc39478832)

[5.8 Reporting Process 36](#_Toc39478833)

[6 Security Services Capability Maturity Index 38](#_Toc39478835)

[6.1 Authentication Capability Maturity 39](#_Toc39478836)

[6.2 Authorization Capability Maturity 39](#_Toc39478837)

[6.3 Boundary Protection Capability Maturity 39](#_Toc39478879)

[6.4 Data Confidentiality Protection Capability Maturity 39](#_Toc39478880)

[6.5 Data Integrity Protection Capability Maturity 40](#_Toc39478881)

[6.6 Injection Prevention Capability Maturity 40](#_Toc39478882)

[6.7 Logging and Alerting Capability Maturity 40](#_Toc39478883)

[6.8 Secret Management Capability Maturity 40](#_Toc39478884)

[6.9 System Integrity Capability Maturity 41](#_Toc39478885)

[6.10 User Session Management Capability Maturity 41](#_Toc39478886)

[7 Security Services Architectural Maturity Index 41](#_Toc39478926)

[Security Service Construction 41](#_Toc39478927)

[7.1 41](#_Toc39478928)

[7.2 Security Service Use 42](#_Toc39478929)

[8 Software Development Maturity Index 43](#_Toc39478975)

[8.1 Accessibility 44](#_Toc39478976)

[8.2 Usability 45](#_Toc39478977)

[9 Documentation Summary 46](#_Toc39478978)

[10 RABET-V Glossary 48](#_Toc39479058)

# The RABET-V Program

This document is for use through the life of the RABET-V Program. This initial version will also inform the RABET-V Pilot Program which is a trial execution of the Program Description. As such, some sections have additional commentary specific to the pilot. These sections are marked in indented italics like the example below.

Example Pilot comments

## Steering Committee

The RABET-V Pilot Program is guided by a steering committee comprised of election officials, election technology providers, and other election infrastructure stakeholders. We will add steering committee member information below as we confirm members:

* Aaron Wilson, Sr. Director of Election Security at The Center for Internet Security (CIS) - Steering Committee Chair
* Jerome Lovato, Testing and Certification Director at The Election Assistance Commission (EAC)
* Don Palmer, EAC Commissioner
* David Beirne, Federal Voting Assistance Program (FVAP)
* Nikki Charlson, Maryland State Board of Elections
* Spencer Wood, Ohio Secretary of State’s Office
* Richard Rydecki, Wisconsin Elections Commission Staff
* Christina Adkins, Texas Secretary of State’s Office
* Jessica Myers, Pennsylvania Secretary of State’s Office
* Mike Moser, Pennsylvania Secretary of State’s Office
* Voting System Technical Oversight Program (VSTOP), Indiana Secretary of State

## Pilot Participants

The following technology providers have volunteered to have their products used in the RABET-V Pilot:

* Scytl - Election Night Reporting
* VR Systems - Electronic Pollbooks and Election Night Reporting
* KNOWink - Electronic Pollbook
* Voting Works – RLA Support Software

## Technology Advisory Committee

The RABET-V Technology Advisory Committee is a growing group of experts in relevant subject matter that are volunteering their time to assist in the refinement of the RABET-V process.

* Jono Spring, SEI CERT Division
* Lauren Cooper, SEI CERT Division
* Brian Glas, OWASP SAMM
* Beau Woods, Atlantic Council
* Mary M Shaw, Carnegie Melon
* David Garlan, Carnegie Melon
* Ryan Wagner, Carnegie Melon
* Joshua Bloch, Carnegie Melon
* Daniel Plakosh, SEI
* Gema Howell, NIST
* Mary Brady, NIST
* Gordon Gillerman, NIST
* Lisa Carnahan, NIST
* Peter A. Bloniarz, New York State Cyber Security Advisory Board
* Rob Gordon

## Program Administration and Research

The program will be administered by CIS team with assistance from The Turnout. Dr. Mike Garcia will serve as the Research Lead.

## Pilot Process

The RABET-V Pilot will first establish a RABET-V Program Description (previously called a Working Model). The Program Description version will detail how each activity will be conducted. The Program Description will be iteratively reviewed by the program Steering Committee and modified, as necessary.

Using the Program Description, the Pilot Program will conduct initial reviews on real products from Pilot Program technology providers. Each initial review will execute all RABET-V activities resulting in the creation of Testing Rules and initial verification results for each product. The Architecture Review and Process Assessments will be conducted according to the architecture and process review steps detailed in the Program Description, which may be updated as necessary throughout the Pilot Program.

The RABET-V process will be conducted on real products from Pilot Program technology providers that represent diverse offerings. The Pilot Program will work with pilot participants to develop their submission package and security claims. This pilot will then follow the architecture and process review technical guidance and develop risk-based product-specific testing processes. The Pilot will evaluate the value of its activities, along with the time and cost, and conclude with recommendations on the best approach.

The Pilot will then conduct multiple iterations of RABET-V on product revisions from the participants. Depending on the changes, RABET-V will adapt and conduct only the activities required. This exercise will highlight the effectiveness of RABET-V to create meaningful but streamlined verifications and help determine the effectiveness of the product architecture and process reviews. It will also provide useful time and cost information. After each RABET-V iteration, changes may be made to the testing process and the iteration repeated as necessary.

## Pilot Research Questions

### Time and Cost Implications

1. What are the review time implications of the RABET-V approach for:
   * The initial verification of a product from a new vendor?
   * The initial verification of a product from a vendor that have been through the RABET-V process?
   * The re-verification of a product?
2. What are the total cost implications of the RABET-V approach for:
   * The initial verification of a product from a new vendor?
   * The initial verification of a product from a vendor that have been through the RABET-V process?
   * The re-verification of a product?
3. Is there a viable economic model for the RABET-V process? If so:
   * Does it require a government agency to drive the program, similar to voting system certification?
   * Is there a model that suppliers in the market can support?
   * Is there a model that states and localities can support?
4. Will the process be efficient enough to keep costs low enough for vendors to make minor updates?

### Market Maturity Implications

1. Is there evidence that products are architected in a manner that is mature enough for the RABET-V process to yield benefits by reducing the extent of re-verification reviews?
   * Will vendors be willing to submit small, frequent updates?
2. Is there evidence that state and local adoption and acceptance processes can leverage the RABET-V process to yield benefits?
   * Can states and localities accept RABET-V verifications quickly enough to make the process worthwhile?
   * Will states and localities be willing to adopt new versions at a rate that maintains incentives to put small, more frequent updates through the process?

### Pre-Review Assessment Implications

1. Is there a sufficient correlation between process assessment results and verification outcomes to use those assessments to expedite verification and re-verification under RABET-V?
2. Should process assessments be renewed and, if so, how often or under what circumstances?
3. What party is best equipped to conduct process assessments?
4. Do architecture reviews provide a sufficient understanding of a given product to determine the impact of:
   * De minimus system changes?
   * Minor system changes?
   * Major system changes?
5. Should architecture reviews be renewed and, if so, how often or under what circumstances?
6. What party is best equipped to conduct architecture reviews?

### Technical Evaluation Implications

1. For which types of non-voting election technology will the process work?
   * Is it better suited for some types of technology over others?
   * How, if at all, does the process have to be modified to make it more suitable?
   * Are vendors more likely to accept the RABET-V process for certain types of equipment?
   * Are states and localities more likely to accept the RABET-V process for certain types of equipment?

# RABET-V Program

RABET-V is a flexible, risk-based, and cost-effective approach to election system verification that will expedite verification of election systems while providing assurances of security and reliability.

The RABET-V Pilot Program is designed to evaluate the RABET-V process and the potential of the process to improve the speed, security assurances, and cost-effectiveness of non-voting election technology verification.

For more information of the Background and Motivation for RABET-V, see CIS’ [How to Improve Election Technology Verification White Paper](file:///C:/Users/wilso/Downloads/Elections_Tech-Ver-White_Paper-2020-0121.pdf).

## Introduction

### Program Goal

The RABET-V Program is a rapid, reliable, and cost-effective approach to verifying the security of non-voting election systems. Its goal is to provide assurances of security and reliability sufficient for stakeholders to have confidence in their use in election administration. Participating organizations will have demonstrated capabilities to effectively build, test, monitor, and maintain their election technology solution through an evidence-based process.

### Program Scope

RABET-V is intended for non-voting election technology systems.

An election technology system is an information system that supports an elections administration process.

A “voting system” is defined in the Help American Vote Act (H.R. 3295, Sec 301) as “(1) the total combination of mechanical, electromechanical, or electronic equipment (including the software, firmware, and documentation required to program, control, and support the equipment) that is used—(A) to define ballots; (B) to cast and count votes;(C) to report or display election results; and (D) to maintain and produce any audit trail information; and (2) the practices and associated documentation used—(A) to identify system components and versions of such components; (B) to test the system during its development and maintenance; (C) to maintain records of system errors and defects; (D) to determine specific system changes to be made to a system after the initial qualification of the system; and (E) to make available any materials to the voter (such as notices, instructions, forms, or paper ballots).”

A non-voting system is any other information system used to administer an election. Examples include voter registration databases, electronic pollbooks, or the websites of government election authorities.

### Definitions

See [RABET-V Glossary](file:///C:\Users\wilso\Downloads\RABET-V_Glossary)

## RABET-V Administrator

The RABET-V Administrator is a central body responsible for overseeing the RABET-V Program. The responsibility includes (some of the terms listed here are described later in this section):

* Accept requests from and manage list of RABET-V Subscribers
* Accept requests from and manage list of RABET-V Registered Technology Providers (RTPs)
* Host and manage content on the RABET-V Public Portal
* Manage the RABET-V Program Description, making changes as necessary and as approved by the Steering Committee
* Execute RABET-V Iterations when Product Submissions are made by RTPs
* Staff or delegate the execution of RABET-V activities with qualified individuals or organizations

The Center for Internet Security, its staff, and contractors, are serving as the RABET-V Administrator for the RABET-V Pilot Program

## Registered Technology Providers

A Registered Technology Provider (RTP) is an organization that develops election technology and has met the minimum requirements in this section.

To be a RABET-V Registered Technology Provider, the technology provider must submit a complete RTP request and agree to the RABET-V Program Commitment. An RTP is responsible for submitting their first Product Submission within 3 months of becoming an RTP.

Registered Technology Providers will be listed on the RABET-V Program Portal.

For the Pilot Program, each pilot participant will be considered an RTP without going through the registration process.

### Registered Technology Provider Request Package

Technology providers must submit a completed request package to become an RTP. A complete package will contain the following information:

* Company Name, Legal Address, and Address(es) of all locations
* Sales and Technical Support Points of Contact
* Website URL
* Company Description

### Program Commitment

RTPs must agree to the RABET-V Program Commitment. The commitment establishes the ethical and responsible behavior expected by all program providers.

The Program Commitment requires:

* Accurate representation of the product capabilities and its security provisions to RABET-V administrators, RABET-V subscribers, customers, and other stakeholders
* Organization implementation and regular assessment of an organizational security framework like the CIS Controls. The RTP should perform organizational security audits regularly and provide the report to the RABET-V Administrator. The report will be provided to RABET-V Subscribers.
* Continuous product maintenance, including the patching of components within reasonable time frames

### Provider Deregistration and Product Delisting

Failure to meet the requirements of the Program Commitment can lead to deregistration of the RTP and delisting of the RTP’s products. Activities subject to deregistration are any that breach the Program Commitment. These include, but are not limited to:

* Inaccurate representation: if the vendor is found to have intentionally mislead RABET-V administrators or its customers as to the capabilities of the organization or the product.
* Lacking organization security: if the vendors fails to subscribe to an organizational security framework, like the CIS controls, and maintain regular audits.
* Lacking product maintenance: if the vendor is no longer properly supporting a product with regular monitoring and maintenance.

### Deregistration Process

TBD

### Delisting Process

TBD

## Subscribers

RABET-V Subscribers are state and local election jurisdictions who intend to use the RABET-V reporting to assist in their certification, approval, or purchase decisions. RABET-V Subscribers will have access to the sensitive information produced by the RABET-V Program. Subscribers must submit a request to the RABET-V Administrator and agree to protect sensitive information. This information will be made available to Subscribers through a secure portion of the RABET-V Public Portal.

An agreement will be drafted for Subscribers to sign prior to being given access.

## RABET-V Public Portal

### Purposes

There will be a RABET-V public portal. The public portal serves the following purposes:

* Lists all RTPs
* Provides a product registry which list all submitted products and the product’s latest goals, expected usage, and security claims
* Lists all Product Versions submitted by RTPs, the date of submission and date of completion, the outcome of the submission, and the RABET-V report

### RABET-V Subscriber Access vs. Public

In an effort to maximize transparency, documentation in the RABET-V process is made as widely available as possible. There are three broad classes of sharing for RABET-V documents and reports:

* Public: documents and reports that are made fully available to the public
* Shared with Subscribers: documents and reports that are made available to those Federal, State, and local election authorities that have requested access and has agreed to treat the information as sensitive
* Shared with the RTP: documents and reports that are shared only with the RTP, though the RTP is free to share documentation with other entities as it sees fit.

# RABET-V Security Services

RABET-V defines ten [Security Services](file:///C:/Users/wilso/RABET-V_Glossary) that are used throughout RABET-V to help evaluate the products:

1. **Authentication:** Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system. [NIST FIPS 200]
2. **Authorization:** The right or a permission that is granted to a system entity to access a system resource. [NIST SP 800-82 Rev. 2]
3. **Injection Prevention:** The rejection or sanitization of data input and output to ensure malicious executable code is not executed.
4. **Key/Secret/Credentials Management:** The activities involving the handling of cryptographic keys and other related security parameters (e.g. passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction. [NIST CNSSI 4009-2015]
5. **User Session Management:** A persistent interaction between a subscriber and an end point. [Adapted from NIST SP 1800-17b]
6. **Logging/Alerting:** The systemic management and monitoring of the events occurring within an organization’s systems and networks. [Adapted from NIST SP 800-92]
7. **Data integrity protection:** Assurance that the data has not been altered in an unauthorized manner. Data integrity covers data in storage, during processing, and while in transit. [Adapted from NIST SP 800-33]
8. **Data confidentiality protection:** Data Confidentiality deals with protecting against the disclosure of information by ensuring that the data is limited to those authorized or by representing the data in such a way that its semantics remain accessible only to those who possess some critical information (e.g., a key for decrypting the enciphered data). [NIST SP 800-13]
9. **Boundary protection:** Monitoring and control of communications at the external boundary of an information system to prevent and detect malicious and other unauthorized communications, through the use of boundary protection devices (e.g. gateways, routers, firewalls, guards, encrypted tunnels). [NIST SP 800-53 Rev. 4]
10. **System integrity protection:** The quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation of the system, whether intentional or accidental. [NIST SP 800-27 Rev. A]

# RABET-V Maturity Indexes

The outputs of the RABET-V process include three indexes that articulate the maturity of the organization and product:

1. [Security Service Capability Maturity (SSCM) Index](file:///C:/Users/wilso/Downloads/MaturityIndexes/Security_Services_Capability_Maturity_Index): scores the product’s current capability across a set of security services. The boundaries of the SSCM are determined by the Security Claims Review activity and the index is generated in the Product Verification activity.
2. [Security Services Architectural Maturity (SSAM) Index](file:///C:/Users/wilso/Downloads/MaturityIndexes/Security_Services_Architectural_Maturity_Index) – measures the maturity and reliability of the product architecture to support each of security services. The SSAM is generated during the Architecture Review activity.
3. [Software Development Maturity (SDM) Index](file:///C:/Users/wilso/Downloads/MaturityIndexes/Software_Development_Maturity_Index) - measures the maturity of the technology provider’s software development processes across a set of security, usability, and accessibility practices. The SDM is generated during the Process Verification activity.

Each maturity index is described in a later section of this document.

# RABET-V Activities

The RABET-V Program consists of eight discrete activities from Product Submission to Reporting. Five of these activities are conditional based on the needs of the submission, scaling or being eliminated based on risks attributed to the product changes. Risk decisions are informed by the product’s architecture, the developer’s processes, and their security claims. Each time the RABET-V process is initiated, it is called a RABET-V Iteration.

1. Provider Submission: A submission from an RTP begins the RABET-V iteration. This submission contains information from the RTP on both its organization and the product under review.
2. Submission Review: The submission is reviewed for completeness and to determine which of the remaining activities are necessary for the submission type.
3. Process Assessment: Assessment of the RTP’s approach to developing software to determine its maturity, which will be used throughout the RABET-V process and subsequent submissions by the RTP. A demonstrably high level of maturity can reduce the burden of review across all activities. One can think of this as assessing the general trustworthiness of an RTP to reliably implement any given product feature or capability.
4. Architecture Review: Assessment of the product’s architectural approach to determine its maturity with regard to various security services. A demonstrably high level of maturity can reduce the burden of review for a specific change. One can think of this as assessing the trustworthiness of the product that changes to one product feature or service will not have security implications for other aspects of the product.
5. Security Claims Validation: Assessment of whether the RTP’s stated security approach are appropriate given the goals and expected use of the product. This assessment results in a set of security requirements that will used in producing the Testing Rules.
6. Testing Rules Determination: Produces a decision tree for testing based on the outputs from the prior activities.
7. Product Verification: Executes the test plan.
8. Reporting: Produces reports and provides results to stakeholders.

A screenshot of a computer

Description automatically generated

Figure 1 - RABET-V Workflow

## Provider Submission

The RABET-V process begins with a product submission from the Registered Technology Provider.

### Submission Types

All product submissions are either an Initial Submission or a Revision Submission.

#### Initial Submission

The Initial Submission is a first-time submission of product information. It includes statements about the product and the RTP that will be used throughout the RABET-V process. An Initial Submission is required for each unique product an RTP would market and sell independently to an election jurisdiction.

#### Revision Submission

A Revision Submission is for changes being made to a product that has already been through the RABET-V process. It includes information about changes to the product since the last submission.

An RTP can make a Revision Submission at any time after that product has been verified through an initial RABET-V iteration. It can improve the likelihood of a smooth process by engaging the RABET-V Administrator ahead of the submission about upcoming changes and understanding how any established Testing Rules will be impacted by deviations from the previous version.

A Revision Submission requires only the change list, artifacts, desired deployment date, and version numbers. It also requires any other submission items that have changed substantially since last going through the RABET-V process.

### Submission Items

#### Product Goals – Initial Submission and On Change

The Product Goals statement is a description of the product’s purpose in non-technical language. It should be brief: a one or two paragraph summary of what the product is designed to do. The RTP can update the Product Goals during any Revision Submission.

This description will be used by the RABET-V Administrator in the Submission Review and Security Claims Review activities to determine if the stated security claims align with the product goals. For example, if the Product Goals include managing sensitive voter information, the RABET-V Administrator will expect to see security claims designed to protect sensitive voter information.

The Product Goals will be published in the [RABET-V Public Portal](file:///C:/Users/wilso/RABET-V_Glossary).

#### Expected Usage – Initial Submission and On Change

The Expected Usage statement describes how the provider expects the election office to use the product. While it can communicate this through a number of means, a good approach is through high-level use cases that list the actions and interactions between involved parties and the system to achieve the Product Goals. Usage of the product will be limited to the use cases expressed in the Expected Usage. The RTP can update the Expected Usage during any Revision Submission.

This description will be used by the RABET-V Administrator in the Security Claims Review activity to determine if the stated security claims align with the expected usage.

The Expected Usage will be published in the [RABET-V Public Portal](file:///C:/Users/wilso/RABET-V_Glossary).

#### Product Security Claims – Initial Submission and On Change

The Product Security Claims statement is a listing of security requirements met by the product. The security requirements are listed as a part of the Security Service Capability Maturity Model and organized by security service.

For each implemented requirement, the provider will describe how they implement it and whether they believe it is Met, Partially Met, Not Met, or Not Applicable. If the provider only implements it on certain components, those should be detailed along with a reason for excluding it from other components. The provider should include well-reasoned arguments for why the implementation decisions were made and how they result in the appropriate level of security for the product. This approach allows each product to implement a unique approach to the security of their application that is specific to its goals and usage.

The RTP can update the Security Claims during any Revision Submission.

#### Process Descriptions – Initial Submission and On Change

The Process Descriptions statement is about the provider’s development and operating environment processes. These should cover key aspects of software development as described in the OWASP Software Assurance Maturity Model (SAMM), which is used as the basis for the RABET-V Process Review Activity.

A lack of detail in the Process Description statement will not exclude a provider from participating in the program, though it may slow the pace of the review.

The pilot program will work with the provider to create the necessary descriptions.

#### Architecture Documentation and Diagrams – Initial Submission and On Change

The Architecture Documentation and Diagrams is a set of documents that describe in full the architectural design of the product. The product’s architecture can be described using diagrams, or a combination of the two.

The RTP should submit documentation of the architecture at the system as well as the software level. The system architecture should describe deployable subsystems, such as web services, databases, as well as hardware components such as firewalls and tablets. The software architecture should be described in terms of software components.

The term *component* is used generically within RABET-V to describe part of a product. Components can be broken down into subcomponents, as required. The architecture should be deconstructed to the level that exposed functionality (e.g. a particular web service, program API, etc.) can be identified.

RABET-V does not dictate a particular notation for diagrams; however, where possible RTP’s should follow provided examples, which are based on UML Component Diagrams.

A lack of architecture documentation and diagrams will not exclude a pilot participant from the program. The pilot program will work with the provider to create documentation and diagrams which are missing.

#### Third-Party Component Details – Initial Submission and On Change

The Third-party Component Details describe the provider’s approach to managing supply chain risk. This includes whether the organization has selected third-party software components with a history of known vulnerabilities, and how the organization maintains traceability and assurance of third-party and open source software throughout the lifetime of the software.

When considering parts of the overall solution that are not developed internally, each unique version of the following will be considered an individual component of the system:

1. Operating System
2. Framework
3. 3rd party API
4. Embedded 3rd Party Library
5. Hosting Software/Service (i.e. IIS, Docker, Elastic Beanstalk, Azure App Service, etc.)
6. Database (database stored functions and procedures will be treated as a part of the software application)
7. File Storage System/Service
8. Network Appliance (virtual or real)
9. External Device Driver/Firmware

A change to one of these components will be treated a change to the entire component and the version number and change list will describe the entire component.

The provider should detail initial and ongoing vetting procedures for third-party providers and components (if not covered in the Process Descriptions), to include open source software and libraries. Vetting should include fit for the provider as well as security and reliability. Management of third parties includes the approach to policies, SLAs, reputation, maintenance, and past performance of third-party software and services.

A lack of documented third-party component details will not exclude a participant from the program’s pilot phase. The pilot program will work with the provider to develop the necessary documentation.

#### Product Environment and User Documentation – Every Submission

The RTP must provide access to a product environment that can be used by the administrator to conduct the RABET-V iteration. This should be a dedicated environment running the new product version. The administrator must provision user accounts and test data consistent with the Expected Usage statement. Test data should not include sensitive information, but may include data from real elections that is sanitized as necessary to remove personal information, product passwords, etc.

On the initial submission, the provider should include user documentation and be available for a training session to assist the administrator in understanding the product usage. Updated documentation and training sessions should be provided when changes are significant enough to warrant the update. User documentation must include the product version number it was written to support.

For many products, the product environment is the deployment of the web application to a sandbox hosting environment. For products like electronic pollbooks with physical devices, the product environment must include deployments of the product revision on physical devices provided to the administrator.

#### Revision Submission Artifacts – Revision Submission Only

A provider can submit a product revision to RABET-V at any time. Engaging the RABET-V Administrator about upcoming changes and consulting the existing Testing Rules will help a technology provider better prepare their submission.

All Revision Submissions require the following artifacts:

1. Change list - Indicates which components have changed and what level of change was made. It should reference the components identified in the architecture review.
2. Artifacts - The product development artifacts identified in the existing Process Review. These artifacts provide the necessary information on product changes to conduct a review of the changes in the Change List
3. Desired Deployment Date - Target date for deploying the product revision in a production environment.
4. Version number - The version number of the current product revision. It must indicate and correspond to code branches and change size (i.e. minor version number changes must correspond to minor changes).

A provider may change any of the Initial Submission items during a Revision Submission by providing updated information and alerting the Administrator.

### Submission

Once the Initial Submission or Revision Submission package is complete, it should be submitted electronically to the RABET-V Administrator.

For the pilot, CIS will provide a method to submit this information.

## Submission Review Process

Once the RTP has made a submission, the RABET-V Administrator will review the submitted information and determine which RABET-V activities are necessary for this iteration.

### Inputs

* Provider’s submission package
* Prior process review, if a revision submission

### Outputs

* Submission Review Checklist indicating submission type, change list, and which RABET-V activities should be performed in this iteration

### Workflow

#### Review package for completion.

See [Provider Submission](file:///C:/Users/wilso/Downloads/Provider_Submission) for submission requirements.

##### Initial submission

All RABET-V activities are required in order to generate the Testing Rules. Ensure all items on the Submission Review Checklist are included in the submission. For each step, indicate on the Submission Review Checklist if the respective item is present or missing.

##### Revision submission

Some RABET-V activities may not be required. Complete the remainder of the steps in this process to determine which activities are required for this submission. For each step, indicate on the Submission Review Checklist if the respective item is present, missing, or not required.

#### Validate change list

The approach to validating the change list will vary based on the findings of the prior Process Review:

1. Reliable: change list validation can be skipped or limited to high level spot checking
2. Otherwise: validate the change list by manual or automated means

Record the result in the Submission Review Checklist.

#### Determine if Process Assessment activity is necessary

The Process Assessment is required when one of the following conditions is true:

1. The submission is an Initial Submission
2. The provider has requested a new Process Assessment in order to generate a new set of Testing Rules or update Software Development Maturity (SDM) scores
3. It has been more than 18 months since the last Process Assessment was performed
4. Artifacts provided by provider indicate a significant process change has occurred within provider

Record the result in the Submission Review Checklist.

#### Determine if Architecture Review activity is necessary

The Architecture Review is required when one of the following conditions is true:

1. The submission is an Initial Submission
2. The provider has requested a new Architecture Review in order to generate a new set of Testing Rules or update Security Services Architectural Maturity (SSAM) scores
3. The change list indicates the addition, removal, or modification of major architectural components since the last Architecture Review

Record the result in the Submission Review Checklist.

#### Determine if Security Claims Validation activity is necessary

The Security Claims Validation activity is required when one of the following conditions is true:

1. The submission is an Initial Submission
2. The provider has updated the product goals, expected usage, or security claims.
3. The provider has requested a new Security Claims Validation in order to generate a new set of Testing Rules or update Security Services Capability Maturity (SSCM) scores
4. The change list indicates that prior Security Claims Validation findings need to be reviewed

Record the result in the Submission Review Checklist.

## Process Assessment

The RABET-V Process Assessment measures the [Software Development Maturity (SDM)](file:///C:/Users/wilso/MaturityIndexes/Software_Development_Maturity_Index) of the technology provider. It uses the OWASP Software Assurance Maturity Model (SAMM) as the basis for its evaluation. This evaluation determines a maturity score for the technology provider in 5 areas across 15 principles. These maturity scores are used to help determine the types of testing conducted by RABET-V for product changes.

In addition to providing the maturity scores, the SAMM evaluation will determine the reliability of provider-generated artifacts that can be used by RABET-V. By using provider-generated artifacts, the RABET-V process will not have to reproduce these artifacts (i.e. test results). The OWASP SAMM evaluation will be conducted by a 3rd party evaluator. OWASP maintains a list of SAMM evaluators. Unless not practical, the SAMM evaluation should be performed by one of these evaluators. These evaluators will review documentation and perform interviews with the technology provider in order to complete the evaluation. Evidence of artifacts – such as historical version of reports – will be required.

The OWASP SAMM project makes a toolkit available. This [toolkit](https://github.com/OWASP/samm/tree/master/Supporting%20Resources/v2.0/toolbox) provides an interview option for evaluating the provider’s processes according to SAMM.

In addition to the SAMM security function, RABET-V adds two additional functions: usability and accessibility. Together, this generates the [Software Development Maturity (SDM)](file:///C:/Users/wilso/MaturityIndexes/Software_Development_Maturity_Index) scores.

### Inputs

* Process descriptions
* Interviews with RTP

### Outcomes

* Software Development Maturity Scores
* List of software development artifacts usable for verification

### Workflow

#### Review Existing Documentation

The provider submitted existing documentation during the Provider Submission activity. This was not a requirement for perfect documentation, just what may already exist. The type of documentation requested includes:

1. Policy and Compliance documents that are related to or help define efforts related to acquiring, managing, designing, developing, testing, and supporting software at the organization.
2. Process related documents that help define which processes are followed related to software activities at the organization.
3. Artifacts from completed activities related to the above policy and compliance or process related activities. Just a representative sample is needed.

#### Discussion Sessions

These sessions are for interactive discussions with the different roles supporting the efforts related to software development at an organization. The discussions will normally involve two interviewers and will last approximately 60-90 minutes. While sessions are driven by topics found in the SAMM toolkit, they will not be checklist-based, but discussions on how processes and procedure are implemented and conducted throughout the organization. Below are some of the common organizational roles that would be interviewed, however representatives from the appropriate business units are also useful candidates for interviewing:

1. Application/Software Security lead or equivalent party with responsibilities for defining and managing the integration of security into software
2. Business Analyst or similar role with responsibilities related to requirements, user stories, etc.
3. Project Manager or similar role with responsibilities for guiding teams through the processes to develop, acquire, and/or maintain software
4. Application Architect or similar role with responsibilities to help ensure good design and architecture for applications is defined
5. Developer or similar role that has responsibilities to write code and some testing
6. QA/Test or similar role that handling the primary testing for software or applications
7. DevOps Engineer or similar role with responsibilities related to build and deployment processes for software
8. Incident response/Support or similar roles with responsibilities for helping support, triage, respond to issues in production systems

#### Determine Artifact Reliability

RABET-V can expedite product verification if certain software development artifacts are found to be reliable. When artifacts are found to be reliable, the RABET-V process may use them instead of reproducing similar tests. However, this does not mean RABET-V must use them. In fact, from time to time, RABET-V may reproduce the results submitted by the provider in order to validate the artifacts are still reliable.

The Process Assessment is used to help determine if the following artifacts are accurate and consistently available for RABET-V iterations. If a technology provider has additional software development artifacts which they believe are reliable and beneficial to streamlining the RABET-V process, they may request those artifacts to be evaluated and the testing rules updated to account for the artifacts.

##### Change List

This is the most important software development artifact used by RABET-V when performing verification of Product Revisions. It is critical that the list is accurate, detailed, and complete. While RTPs can submit manually generated change lists, they may take longer to process than automated change lists built from the central source code repository and reviewed by system architects and product owners.

During the Process Assessment, the method used for building change lists will be discovered and sample change lists will be reviewed for accuracy and completeness. If the change list is determined to be reliable, the RABET-V process will use the provider’s change list and not generate its own. If the change list is not reliable, the RABET-V process will explore other ways to produce an accurate change list - which may take additional time and resources.

##### Automated Configuration Assessments

Security configurations are a major part of ensuring that systems contain properly implementing security controls. Using configuration guidance, such as the CIS Benchmarks, leads to consistent security outcomes. Using automated configuration assessment tools, such as the CIS configuration assessment tool (CIS-CAT) can ensure guidance is being followed for every release.

During the Process Assessment, the reviewer will determine if the technology provider is subscribed to configuration guidance and if they are using a reliable assessment tool. If so, the results of the assessment tool will be used during RABET-V iterations to verify certain requirements. If this artifact is not present or reliable, the Product Verification activity will have to perform additional testing to verify secure configurations.

##### Automated Vulnerability Assessments

Automated vulnerability assessments check system components for known vulnerabilities. These assessments primarily check third party components for known vulnerable versions of software. RTPs who are found to be regularly performing automated vulnerability scans on the product networks and software will have their results used during the Product Verification activity in lieu of RABET-V reviewer performing new scans. During the Process Assessment, reviewers will investigate the scope, frequency, and tooling used by the technology provider to determine if there is sufficient coverage and accuracy.

##### Automated Unit Testing

Automated unit testing is a way to regression test large and complex applications efficiently. It takes significant investment on the part of the technology provider to build test suites that are robust and accurate. For RTPs that have invested in this capability, the results of their internally testing can be used to offset of the RABET-V verification. The Process Assessment will look at the coverage and depth of the current automated testing routines, as well as the RTP’s commitment to maintaining their test suites.

##### 3rd Party Security Analysis

RABET-V strongly encourages RTPs to receive regular, in-depth security audits on their systems. For example, there are audits that focus on hosting security and application security. These audits, if performed against a reliable standard and performed recently, can be used in RABET-V in lieu of repeating similar evaluations.

#### Analysis and Reporting

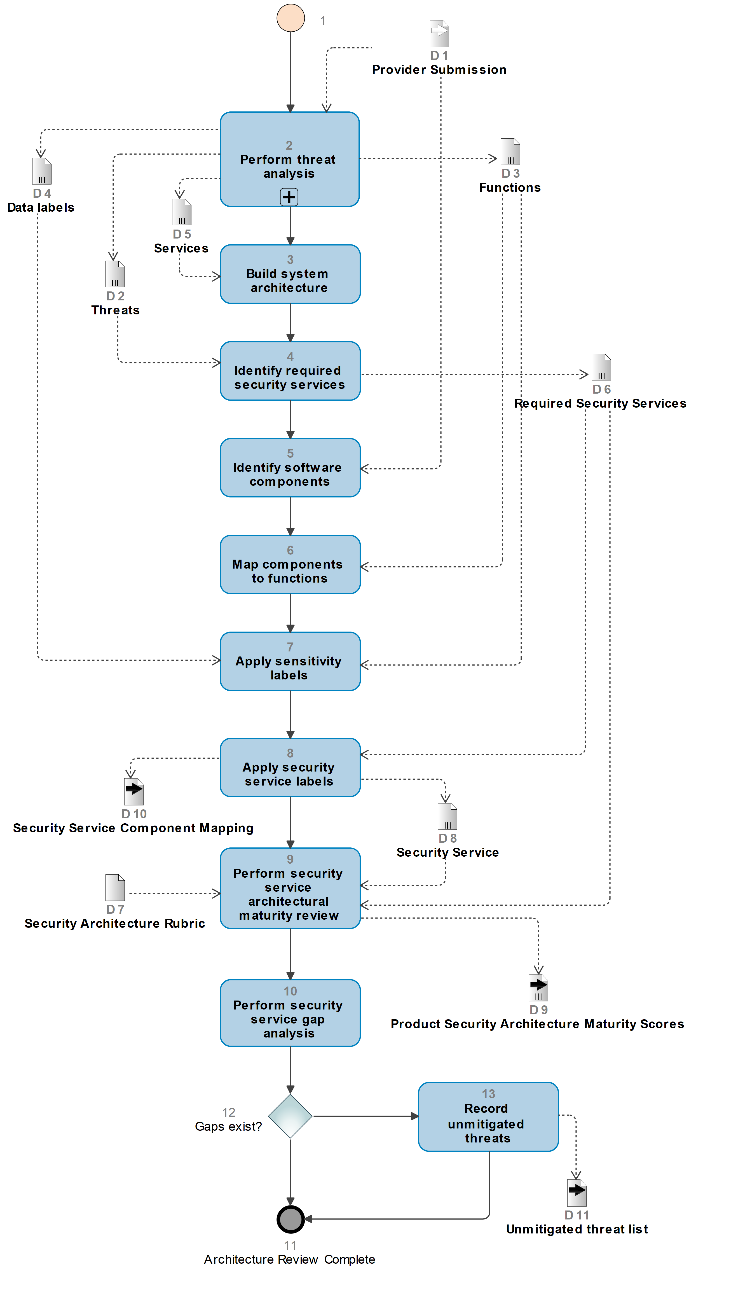
Analysis of the provided documentation (if any) along with the captured session notes will be used to complete a SAMM assessment for the organization. At the conclusion of the analysis, the following artifacts will be delivered as part of the work product for the organization:

1. High level executive summary of the process, findings, SDM maturity score and tailored recommendations
2. Completed SAMM Toolbox
3. Interview session notes

### Technical Guidance

1. [OWASP SAMM](https://owaspsamm.org/)
2. [NIST Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF)](https://csrc.nist.gov/publications/detail/white-paper/2019/06/11/mitigating-risk-of-software-vulnerabilities-with-ssdf/draft)
3. [Managing Security Risks Inherent in the Use of Third-party Components](https://safecode.org/wp-content/uploads/2017/05/SAFECode_TPC_Whitepaper.pdf)

## Architecture Review Methodology

The RABET-V Architecture Review is designed to evaluate the solution’s architectural support for the [RABET-V security services](file:///C:/Users/wilso/MaturityIndexes/Security_Services). This evaluation produces an architectural maturity score for each security service and identifies the components which provide the security service. This score does not measure how well the product executes the security service (i.e., its capability level), just how mature the architecture is that supports the security service. The [Security Services Capability Maturity](file:///C:/Users/wilso/MaturityIndexes/Security_Services_Capability_Maturity_Index) level is a separate metric determined in the [Security Claims Validation](file:///C:/Users/wilso/Downloads/Security_Claims_Validation) and it indicates how well the product provides the security services.

The Architectural Maturity scores and component mappings are used to help assess the risk that changes to the product will negatively impact the security services. These are used in the [Testing Rules Determination Activity](file:///C:/Users/wilso/Downloads/Testing_Rules_Determination) to identify how to test the product changes. The higher the architectural maturity scores, the less testing required to validate the security capability scores.

The Architecture Review is supplied with architecture diagrams, architecture descriptions, and access to a functioning version of the solution. The architecture review will use the functioning solution to validate or fill-in missing pieces from the architecture diagrams and descriptions.

Software code may be requested in the pilot in order to determine if it significantly improves the reliability of the architecture review.

For more information about what is expected for the architecture diagrams and description, see the [Provider Submission](file:///C:/Users/wilso/Downloads/Provider_Submission) activity.

For each security service, the Architecture Review will identify the product components at the system and software levels that *PROVIDE* and *CONFIGURE* the security service and those components that *USE* the component that provides the security service. The components which *PROVIDE* or *CONFIGURE* the security service are referred to as [1st Degree components](file:///C:/Users/wilso/RABET-V_Glossary). The ones which use the 1st degree components are referred to as [2nd Degree components](file:///C:/Users/wilso/RABET-V_Glossary). The 1st and 2nd degree components together comprise the Security Service Architecture.

This activity will also complete the system, security and software architectural viewpoints. The system level diagram identifies the larger components of the environment used to host and manage the election technology software application(s). The software level diagram identifies the components a layer deeper into the election technology software application(s).

### Inputs

* Provider Submission

### Outputs

* SSAM Scores
* Security Service Architecture
* List of unmitigated threats (if any)

### Workflow

#### Perform threat analysis

Threat modeling and analysis is used to build the security architecture viewpoint. It also aids in the development of the system and software architecture diagrams.

Threat modeling takes the provider submitted architectural documentation as input. The services provided by the application are enumerated using the threat modeling methodology. The services are then further deconstructed into software functions and the data required to perform those functions. The data flows/functions must be annotated with sensitivity labels (data-critical, data-sensitive) and impact (low, medium, high) which will influence the severity level determination of any identified threat. Automated tooling will perform threat analysis on this system representation.

RABET-V will evaluate the use the [Microsoft Threat Modeling tool](https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling) during pilot.

#### Build system architecture diagram

The output of the threat analysis is used to build out a system level architecture diagram. This is done by converting the top-level services (e.g. Web Applications, Virtual Machines) and data stores (e.g. SQL Server, MySQL) into subsystem-level components.

#### Identify required security services

The security services required for a given product will depend on the results of the threat modeling exercise. A subset of security services are selected from the 10 security services identified for RABET-V. This is accomplished by reviewing the results of the threat modeling exercise and identifying the security services that mitigate one or more identified threats.

#### Identify software components

While the threat analysis helps identify the system level components, the software level components must be identified separately. This step will identify significant software architectural components, their boundaries, how they interface, and their dependencies with one another and 3rd party components.

Automated tools, such as [Lattix](https://www.lattix.com/), will be evaluated in the RABET-V pilot as a way to perform this step.

#### Map components to functions

The threat analysis identifies the software functions that need security services and the previous step identified the software components. This step will map the security services to software components by first identifying which components offer which functionalities. This is currently a manual process but may be assisted with architectural analysis tools.

#### Apply sensitivity labels

Sensitivity labels are applied to parts of the architecture that provide/exchange data-critical or data-sensitive information during the threat analysis step. Using the component to functionality relationships identified in the previous step, we map and apply the sensitivity labels to components in this step. These labels are used to identify gaps in security service protection in a later step.

#### Apply security service labels

In this step, we complete the Security Service Architecture by labeling the system and software components, for each security service, that provide or configure the security service.

#### Perform security service architectural maturity review

Apply the security service architectural maturity rubric and assign a score to each identified security service.

For more information please see [Security Service Architectural Maturity](file:///C:/Users/wilso/MaturityIndexes/Security_Services_Architectural_Maturity_Index)

#### Perform security service gap analysis

Analyze the architecture and identify if any sensitive interface(s) are not protected by a security service. This could be due to a missing or incorrectly configured security service.

#### Record unmitigated threats

Record the unmitigated threats in the product.

### Data used in Process

#### Threats

A threat is type of a situation that may lead to one or more related incidents or failures. The threat consists of the existence of zero or more threat actors together with a set of one or more vulnerabilities. Thus, the threat of theft may result in an actual theft (attack), and threats correspond to attacks that are typically classified by attacker motivation (e.g., financial, political) as opposed to technique (e.g., spoofing).

#### Data labels

A sensitivity label applied to data. Two data labels are defined for RABET-V:

1. Data Criticality - a label indicating the sensitivity of the data the component is handling. This may be thought of as a label of “integrity”. This is measured by the impact of the data being manipulated to an unknown or incorrect value. Criticality can be determined by examining a component’s exposed interfaces.
2. Data Sensitivity - a label indicating the sensitivity of the data the component is handling. This may be thought of as a label of “confidentiality”. This is measured by the impact of the data being exposed to an unauthorized party. Sensitivity can be determined by examining a component’s exposed interfaces.

For more information, see the related definition for [Data Sensitivity](file:///C:/Users/wilso/RABET-V_Glossary.md#Data-Sensitivity)

#### Services

A system level component that provides data processing capabilities.

For more information, see the related definition for [Process](file:///C:/Users/wilso/RABET-V_Glossary.md#Process)

## Security Claims Validation

This security claims validation activity reviews whether the technology provider’s statements of security are sufficient for their product’s goals and expected usage. Not all applications pose the same security risks. Even similar products can have different risk profiles based on the type of data they manage and how the product will be used. This activity reviews the particulars of the product to ensure the security claims match its specific risk profile. This validation activity determines if the provider’s claims make sense given the product environment and data sensitivity, and if the claims are sufficient for the given context.

Security claims are submitted by providers in their submission package. The Initial Provider Submission must include claims for each security requirement. Subsequent Revision Submissions can add, remove, or modify a previous security claim, as necessary.

For each requirement, the provider must include:

1. Whether the requirement is 1. Met, 2. Partially Met, 3. Not Met, or 4. Not Applicable
2. Which component or sub-systems implements the requirement, and whether it is all or a sub-set of components
3. Rationale for Not Applicable (only if Not Applicable is claimed)
4. Implementation details
5. Explanation for why the requirement is only partially met or partially applied to the system. Simple explanations can be used (i.e. planned for future development, lack of resources, etc.) If the provider believes that partial implementation is sufficient, a longer explanation is necessary.

The Product Verification activity performs the verification of these claims, based on the Testing Rules created in the Testing Rules Determination activity. The Testing Rules Determination accounts for the security claims made by the provider. For instance, the testing rules will exclude requirements which are Not Applicable.

### Inputs

* Product goals (included in provider’s submission package)
* Product expected usage (included in provider’s submission package)
* Product security claims (included in provider’s submission package)
* Product demonstration or access
* Security Service Component Mapping (from architecture review)

### Outputs

* Validation or rejection of security claims sufficiency
* List of applicable security requirements

### Workflow

#### Review Product Goals, Expected Usage, and Product Functionality

This first step will review the written goals and usage from the provider. This step should be augmented with a product demo or access to the product in a test environment. Reviewers should obtain a good sense of the high-level product functionality and validate the goals and usage and consistent with the product functionality. For example, if there are use cases related to product administration, the reviewer should be able to access the administration module and exercise a few use cases.

#### Review Requirements listed as Not Applicable

The requirements marked Not Applicable are reviewed to ensure that for this product the requirement is not relevant and thus the SSCM scores should not reflect non-conformance in its numbers. This is done with the aid of the threat analysis and security service component mapping from architecture review. Using the component mapping and knowledge from the product demo and expected usage, the reviewer should be able to make a determination on whether or not the requirement is valid for this product. Often times, the decision comes down to the use of certain technologies in the system. For example, if the product disabled all wireless, the requirements on using encrypted wireless are N/A.

#### Review Remaining Requirements

Once the list of applicable requirements has been determined in the prior step, the reviewer will go through the applicable requirements one security service at a time. Using the security service component mapping and the implementation details, the reviewer can validate if the stated implementation is fully, partially, or not meeting the requirement. Requirements are fully met when the implementation of the requirement covers all the relevant components. If the implementation is only covering a portion of the relevant components, the requirement is partially met. The determination of which components are relevant is made by the reviewer.

#### Determine Claim Sufficiency

In this final step, the reviewer will analyze the product’s use cases, the list of applicable requirements, and the provider’s validated claims. If there are sensitive use cases which are not being mitigated to a minimally acceptable level, the reviewer may determine that the claims are not sufficient.

Until more guidance can be developed on what is minimally acceptable, the guidance is that any product which claims to meet all applicable maturity level 1 requirements will pass this step.

### Risk Considerations

When determining whether requirements are applicable and which components are relevant, the following considerations are used to help determine risk. This is not an exhaustive list.

#### Data Sensitivity

For this consideration, we review the data that the system processes and how critical the data is to make available (availability), keep confidential (confidentiality), or keep authoritative (integrity). The following sections review some of the most critical election data elements, identify the typical lifecycle for that data, and discuss the points where its value is the highest.

##### Jurisdictional

Jurisdictional data is persistent and defines an election jurisdiction. This includes information about the structure of the jurisdiction such as districts, precincts, and offices. This information is often contained within the county voter registration or election management system and shared with other systems that consume it. When elections are built, a snapshot of this information is used to help define the election. This data is key to building correct ballots and ensuring that voters can vote for the correct candidates and issues. The jurisdictional data used to assign ballot contents to the correct geographical districts and polling places is most at risk when it is used to build the ballots and assign them. This puts the entire supply chain of that data leading up to ballot generation at risk.

##### Voter

Voter information is persistent data that identifies eligible voters and their attributes to help the jurisdiction interact with each voter. This data is managed in the state or locality voter registration systems. Depending on the state, the data flow between state and locality voter registration systems varies with the consistent goal of keeping both systems in sync. Voter data is then consumed by various software applications that interact with voters. This includes electronic pollbooks, electronic ballot delivery systems, online voter registration portals, sample ballot lookup portals, and others. Only certain fields—not the entire voter record—are shared with these other software applications.

Voter information has full, filtered, and public versions. In most states, registering to vote requires the sharing of personal information such as Social Security number (SSN) and driver’s license number. These data elements combine with the voter name, address(es), and other attributes to constitute the full version of the voter record. Most states consider voter registration records to be public, but some of the sensitive information such as SSNs are redacted. This version is known as the public version. Various other filtered formats are available for products that use voter information. For example, electronic pollbooks may require voter records with the driver’s license number but not the SSN. Distinguishing between these versions is important when discussing risks.

Full voter records are at risk throughout their lifecycle due to their value in perpetuating identity theft. They are also in danger of manipulation in the voter registration system because it is the source of voter records for all other systems. Other forms of voter records used for voter eligibility are at most risk of manipulation or deletion when they are transferred and used in electronic ballot delivery or electronic pollbooks. Even though certain sensitive data fields are removed, the data is still extremely valuable for an attacker who may wish to alter a voter’s eligibility to vote or impact for whom they can vote. While these downstream uses of voter records could always refresh themselves from the voter registration database, a well-timed attack on the local version of the voter records could be very impactful.

##### Election

Election data is a combination of jurisdictional information, candidate filing information, and other attributes. Elections are defined by the offices and issues that will appear on the ballot along with their eligible candidates and options. Though there is no definitive composition of an election definition, many refer to the election data as the election definition. Each technology implementation will have a unique specification for an election definition.

Typically, the Election Management System (EMS) creates the election by combining disparate pieces from multiple systems. The term EMS is itself used differently in various places. In some instances, the EMS is an online system often embedded in the voting registration system where the state and counties collaborate to define the election or parts of the election.

In other instances, there is a component of the county voting system that is called the EMS. This is the part of the voting system that finalizes the election and builds the ballots. In all cases, the term refers to a system that contributes to the definition of the election in part or in whole. Also, consistently, the EMS is responsible for communicating the election definition with various consumers. These consumers include public communication channels, ballot marking devices, ballot tabulators, and ballot printers.

Election definitions created by voting systems typically define the construction of the ballots and the rules by which poll workers and voters interact with the ballots. In some cases, the election definition may also include configuration data for election security—keys, passwords, PINs, etc.—and how to tabulate ballots. The election definition is what is used to program the various technology components of a voting system. Once the election definition is created and approved by the jurisdiction, its integrity has critical value. Modifications to the election definition can significantly alter how the election is conducted and the outcome of the process.

##### Ballot (blank)

Ballot data is the collection of ballot contents into ballot styles and may take the form of structured data or printable forms like PDFs. Ballot data is a subset of election data but is separated for our purposes because you will often find ballot data isolated from election data. When isolated from election definitions, ballot data has a unique risk profile. Ballot data is often created by the voting system’s election management function and is then distributed to various consumers. This includes ballot printing companies, on-demand ballot printers, ballot marking devices, and electronic ballot delivery systems.

Modification of blank ballots can disenfranchise voters or manipulate how their intent will be read by the voting machine tabulators. For example, a blank ballot could be altered to switch the order of candidates. The election definition is programmed to read the ovals in the original order, but the voter marks the ovals according to what they see on their ballot. This will cause their vote to be attributed to the wrong candidate. Blank ballots are most at risk from the time they are approved by the election jurisdiction to when they are presented to the voter for marking.

##### Election Results

Election results are the aggregated totals generated from voting system tabulation functions. These typically come in summary and detailed versions. The most common detailed version is precinct-level results, but this may also refer to results by district. Election results are generated by the voting system in various individual machines and then aggregated into a central result reporting function of the voting system. Election results are initially considered “unofficial” and then go through a canvassing process that will certify the results as official. The canvassing process differs based on state law and by office.

Most people recognize the lifecycle of election results as beginning when the polls close on election night. In fact, the lifecycle begins when the first ballot is cast in the election, which may be weeks before Election Day. As soon as the first ballot is scanned, the tabulator will store results including a ballot image in most systems. The results are maintained and updated on individual machines until they are aggregated by the election jurisdiction. Typically, the election jurisdiction will wait until polls close on election night to aggregate results from the individual tabulators into a results reporting system. This can be done one of three ways. First, the results can be manually entered from results tapes. Second, the removable media from the tabulator can transfer the results to the results reporting system. Third, the results may be remotely transferred from the tabulator. The last option is only available in some states and is only used for tabulators used on Election Day.

Once an election jurisdiction aggregates the results it has when polls close on election night, those results are transferred from the voting system to an election night reporting solution. The voting system is typically offline, and the election night reporting solution is an online system. This transfer is typically done using USB-based removable media. Once the results are on an election night reporting solution, they are made available to the public using an election night reporting website. For the most part, the risk to election results is the risk to their integrity. However, it is equally important to protect the confidentiality of election results prior to polls closing.

Election night results are a form of unofficial election results. Those results are special because they don’t go through rigorous review, are stored and displayed from internet-connected web servers, and are sometimes aggregated from results sent by vote tabulators over public networks. Nevertheless, they are immediately trusted by the public. As such, these results are at significant risk of tampering and manipulation. The outcome of such tampering would lead to widespread confusion and distrust in the correct result produced by the voting system.

#### Election Operations Criticality

This consideration reviews how critical the product, and its services, are to an election’s operations. Is the product a single point of failure? What options are available as backups for election officials? Is the product used during non-peak times or peak times?

## Testing Rules Determination

This activity takes the results from previous activities and builds a unique set of Testing Rules for each product. These Testing Rules stay valid as long as none of the previous activities - Architecture Review, Process Assessment, and Security Claims Validation - need to be redone. If any of those activities are performed during the current RABET-V Iteration, the Testing Rules Determination must be performed again.

The Testing Rules are structured as a decision tree. A decision tree is a flowchart-like structure where each internal node is a “test” of a condition. Each branch represents an outcome of the test until a final node has been reached. The final node represents a decision.

In our application of decision trees, product changes are the inputs to the decision tree which are then taken through a series of questions about the change and the result is a verification method to use. The decision tree is different for each product because the master tree is simplified to only branches that are relevant for the product and provider.

### Inputs

* Applicable Security Claims
* Security Service Architectural Maturity Scores
* Software Development Maturity Scores

### Outputs

* Testing Rules Decision Tree

### Workflow

#### Initial Decision Tree

Any time the testing rules need to be created or updated, start with a copy of the master decision tree. The master decision tree has all possible outcomes, even if they are not relevant for the product.

The decision tree is currently implemented with [Silver Decisions](http://silverdecisions.pl/). The master decision tree is located in this repository at [master decision tree](file:///C:/Users/wilso/Downloads/Decision_Trees/decisiontree_master.json).

#### Simplify Master Tree

The master decision tree has a number of test conditions for maturity scores which are known at this point in the process, and thus can be simplified. To simply, remove the branches of the decision tree which are irrelevant. This will be branches corresponding to maturity scores other than the score the product/provider received. This should leave a simplified decision tree.

### Test Conditions

#### Change Attributes

These test conditions are related to where the change was made and what kind of change it is.

##### Change Location

Where is the change?

1. Software Application Core Component - custom application developed by technology provider
2. Software Application Supporting Component – a component used from within the provider’s software application such as a framework, library or module provided by a 3rd party
3. Supporting Environment - system, framework, or service provided by a 3rd party used to host or run the software application

##### Change Type

What type of change is it?

1. Add/Remove - a component was added or removed
2. Replace - the component was replaced
3. Interface change - the component interface was changed or changed how it interfaced with other components
4. Implementation/Patch -

* For internally developed components, the internal implementation of a component was changed without impacting the interface(s)
* For 3rd party components patch - the component was patched by its provider (only applies to 3rd party components)

Since more than one may apply, the first one that applies should be selected.

#### Architectural Considerations

These test conditions relate to how the change impacts security services.

##### Security Service Relation

Was the change in a security service related component?

1. 1st Degree Security Service component - the change was in a 1st degree security service component (1st and 2nd degree security service components are determined in the architecture review)
2. 2nd Degree Security Service component - the changes was in a 2nd degree security service component (1st and 2nd degree security service components are determined in the architecture review)
3. None

##### Security Service Architectural Maturity

What is the architectural maturity of the security service impacted?

This uses the Security Service Architectural Maturity (SSAM) scores from the architecture review. Uses 3 options to indicate the level of maturity. The scores ranges below are just placeholders until more accurate score ranges can be determined.

* Initial understanding and adhoc provisions: Less than 1.0
* Increased efficiency and/or effectiveness: At least 1.0 but less than 2.0
* Comprehensive mastery: At least 2.0

#### Software Development Considerations

These test conditions are related to the technology providers Software Development Maturity (SDM) scores and artifacts.

##### Software Assurance Maturity

How mature are the provider’s processes related to software assurance?

This uses the average score of all SAMM principles with 3 ranges. The scores ranges below are just placeholders until more accurate score ranges can be determined.

* Initial understanding and adhoc provisions: Less than 1.0
* Increased efficiency and/or effectiveness: At least 1.0 but less than 2.0
* Comprehensive mastery: At least 2.0

##### 3rd Party Component Maturity

How mature is the technology provider process for vetting 3rd party components?

This uses the SAMM streams of Supplier Security and Software Dependencies with 3 ranges. The scores ranges below are just placeholders until more accurate score ranges can be determined.

* Initial understanding and adhoc provisions: Less than 1.0
* Increased efficiency and/or effectiveness: At least 1.0 but less than 2.0
* Comprehensive mastery: At least 2.0

##### Internal Development and Deployment Maturity

How mature are the technology providers processes for internal development and deployment?

This uses the SAMM streams of Security Requirements, Build Process, Software Dependencies, Deployment Process, and Secret Management with 3 ranges. The scores ranges below are just placeholders until more accurate score ranges can be determined.

* Initial understanding and adhoc provisions: Less than 1.0
* Increased efficiency and/or effectiveness: At least 1.0 but less than 2.0
* Comprehensive mastery: At least 2.0

##### Environment Management Maturity

How mature are the technology provider’s processes for environment management (i.e. configuration hardening, patching, and updating)?

This uses the SAMM Environment Management score with 3 ranges. The scores ranges below are just placeholders until more accurate score ranges can be determined.

* Initial understanding and adhoc provisions: Less than 1.0
* Increased efficiency and/or effectiveness: At least 1.0 but less than 2.0
* Comprehensive mastery: At least 2.0

##### Software Development Artifacts

The following test conditions look at whether reliable artifacts are available.

Is there a reliable artifact available? Examples include:

* Automated source code unit test results
* Automated vulnerability test results
* Manual testing artifacts (test run results, example outputs, etc)
* Automated configuration verification results
* Security event audit logs
* 3rd Party security analysis results (automated or manual)

## Product Verification Activity

The purpose of the product verification activity is to finalize the Security Services Capability Maturity (SSCM) scores for this product revision. For some product changes, this activity will be streamlined because the changes were determined to pose a low risk to the current security capability scores. For other changes, this activity will be extensive in order to determine, or redetermine, the proper maturity scores. The risk is determined in the Testing Rules which produces a Test Plan commensurate with the risk.

### Inputs

* Testing Rules
* Product Revision Submission (Product Revision deployed to test environment, Product development artifacts)
* Component definitions from Architecture Review (used to set scope of testing)

### Outputs

* Results of verification test methods

### Workflow

#### Test Plan Generation

The Test Plan for the Product Verification activity is generated from the product’s Testing Rules. The Testing Rules are built in the Testing Rules Determination activity and may be recently created or be existing rules from prior RABET-V iterations.

The Testing Rules are a decision tree where each change is processed by the tree and the end result is a verification method(s) to use. This must be done for all changes and the Test Plan is the aggregation of all verification methods.

For initial submissions, a full system test is performed. A full system test is will review automated test results, perform a system wide functional test and penetration test.

#### Execute Test Plan for Security Requirements

The Test Plan will identify how to test the product revision using one or more of the verification methods. Each verification method has its own workflow.

#### Sanity Testing for Product Type Requirements

RABET-V is primarily a security verification process. However, it is critical that each product revision processed by RABET-V meet basic product requirements for its stated purpose. These basic product requirements will vary by product type and are managed separately from the RABET-V Program Description.

During the initial RABET-V iteration, sanity testing (testing that is done to ensure that all the major and vital functionalities are working correctly) will be performed against this basic set of product requirements based on the product type. For subsequent RABET-V iterations, the testing rules will indicate whether sanity testing is necessary and whether it is limited or full. Limited sanity testing is focused on the changed component. Full sanity testing will perform testing on all requirements.

### Verification Methods

As indicated in the Test Plan, the RABET-V administrator, or its designee, will use one of more of the following techniques. The scope of the testing (i.e. which components to test) will also be indicated by the Test Plan.

#### Artifact Review

This method will review an artifact provided by the technology provider. The review will look for gaps or concerns in relevant security controls based on the information provided. Each type of artifact will have various indicators of acceptability. Types of provider artifacts include:

* Automated source code unit test results
* Automated vulnerability test results
* Automated configuration verification results
* Security event audit logs
* 3rd Party security analysis results (automated or manual)

The artifacts must be evaluated as “reliable” during the Process Assessment activity in order to be used for Product Verification.

#### Automated Testing

Automated testing is broad type of testing relying on software to perform test routines against the product or product component. Automated testing will execute the testing software against its target and produce results which will be evaluated by the RABET-V Administrator or its agent. The type of automated test will depend on the target. The types of automated testing currently conceived for RABET-V include:

* Configuration Testing
* Vulnerability Analysis
* Source Code Analysis
* Accessibility Testing
* Browser Compatibility Testing

#### Functional Testing

Functional testing is a broad type of testing which focuses on the system output (i.e. the functionality users can interact with). It is geared toward testing the functional requirements of the product and is a manual testing method. The scope and intensity of functional testing can vary, and there are sub-types of functional testing to indicate the scope and intensity. The following sub-types are used in RABET-V:

* Component - testing which evaluates a singular component and the requirements associated with that component
* Sanity - testing that is done to ensure that all the major and vital functionalities are working correctly
* Regression - testing performed to ensure that adding new code, enhancements, fixing of bugs is not breaking the existing functionality or causing any instability and still works according to the specifications.
* Integration - validation that multiple components work coherently when operating together.
* System/Sub-system - testing that is performed on a complete system or sub-system to verify it works as expected once all the modules or components are integrated.
* End to End - testing performed to verify the functionality of the product.
* Exploratory/Ad-hoc - informal testing to explore the application and looks for defects which exist in the application.

#### Penetration Testing

Penetration tests evaluate the product to find security vulnerabilities that an attacker could exploit. The scope of a penetration test may be the product’s network, computer systems, or software application(s). In RABET-V, the penetration testing is limited to web application penetration testing. Web applications expose the greatest surface area for which automated testing is incapable of fully evaluating. Automated tools are fairly effective for network and computer systems where the major issues are vulnerable versions and lack of patching. Web applications, however, are custom and may have a variety of issues not easily captured by automated tools. Automated tools help with web application pen tests but must be used by skilled and experienced testers.

RABET-V relies on the [OWASP Web Application Security Testing Guide](https://github.com/OWASP/wstg/tree/master/document/4-Web_Application_Security_Testing) to segment up the penetration testing options.

In addition to a full penetration testing option, the following web application penetration testing subtypes are supported:

* Configuration and Deployment
* Identity Management
* Authentication
* Authorization
* Session Management
* Input Validation
* Error Handling
* Cryptography

Limited penetration testing may be used if the changes do not warrant full penetration testing.

### Out-of-scope Testing

There is other testing which is out of scope for RABET-V. RABET-V is chiefly concerned with verifying the security and reliability of the product revision in a rapid way. Since rapid change cycles are possible with RABET-V, other user-centered types of testing can be performed by the current or potential end users and the changes reprocessed through RABET-V without significant lag. RABET-V reports can be used by state authorities or state and local users to determine the level of this testing necessary. These other testing types include:

#### Acceptance Testing

Acceptance Testing, or User Acceptance Testing (UAT), is performed by the client and verifies whether the end to end flow of the system meets their business requirements or not. The client accepts the system only when all the features and functionalities work as expected.

#### Beta Testing

Beta Testing is carried out by the customer or potential customer. It is performed in the real environment before releasing the product to the market for the actual end-users. Beta Testing is often used to ensure that there are no major feature gaps or bugs in the product, and it satisfies the business requirements. Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.

#### Usability Testing

Under Usability Testing, the user-friendliness is verified. The application flow is tested to know if a new user can understand the application easily or not and if proper help documentation is provided. RABET-V measures the provider’s usability and accessibility testing process maturity, but the ultimate usability testing should be performed by the end-users.

## Reporting Process

### Inputs

* Results from Product Verification activity

### Outputs

* Decision (see Decision Types)
* RABET-V Product Provider Report
* RABET-V Product Public Report

### Workflow

#### Review of Product Verification Results and Determination

An internal review of the Product Verification Results will examine whether the product’s verification met its claims.

The internal review will result in a Verification Status. The possible Verification Statuses are Verified, Conditional Verified, and Returned. These determinations are published in the Public Portal and may be updated if a Verification Status changes, most commonly when a Conditional Verified product has made adjustments that move it to a Verified status.

##### Verified

A Verified status means that the product is likely to perform as described in its Product Goals, and Security Claims in the Expected Usage operating environment.

##### Conditional Verified

A Conditional Verified status means that while the product is likely to perform as described in its Product Goals and Security Claims in the Expected Usage operating environment, the RABET-V process identified at least one non-critical issue or deviation.

With a Conditional Verification, the provider is expected to remediate the issue for a re-verification. If no other changes are made to the product, the process for re-verifying is considered part of the same submission and, upon review, can result in the Verification Status being changed to Verified.

Issues and deviations are detailed in the Product Provider Report.

##### Returned

A Returned status means that the product does not perform as described in in its Product Goals and Security Claims. It has critical issues or deviations that are unlikely to be addressed through minor fixes. The RABET-V process identified at least one critical issue or deviation, severe enough that additional review will require a new submission.

Issues and deviations are detailed in the Product Provider Report.

#### 2. Product Provider Report Generation

##### Report Template

The RABET-V Results Summary provides scored outcomes for product security capabilities and security architecture maturity and for organizational software development process maturity. For Revision Submissions, it will include any change from the previous submission.

Product Security Capability Maturity: the quality of the product’s capabilities of the system at providing each of these security services:

* Authentication
* Authorization
* Injection Prevention
* Key/Secret/Credentials Management
* User Session Management
* Logging/Alerting
* Data integrity protection
* Data confidentiality protection

Product Security Architecture Maturity: the quality and reliability of the product’s architecture to support these security services and the likelihood that product changes will impact the Product Security Capability Maturity levels:

* Authentication
* Authorization
* Injection Prevention
* Key/Secret/Credentials Management
* User Session Management
* Logging/Alerting
* Data integrity protection
* Data confidentiality protection

Software Development Maturity: the quality of the provider’s processes in each of these areas:

* Governance
* Design
* Implementation
* Verification
* Operations
* Usability

Product (Revision) Summary

* Details about the product that were submitted including its description, expected usage (i.e. use cases), version number(s), etc. This includes the Change List for Revision Submissions.

Verification Methods

* Description of how the system was tested to include verification methods used in the testing.

Maturity Trends

* A description of what caused a change for any product or process maturity level that changed.

Appendices

* Requirements Scores: a list of all individual requirements and whether the provider is meeting them

#### 2. Product Public Report Generation

Each completed Verification will have a public report that provides basic information on the verification. This information will include:

* A reference number for the review
* The product’s name and version number
* The provider’s name
* The initial Verification Status and date
* The current Verification Status and date
* Contact information for the provider

# Security Services Capability Maturity Index

The SSCM Index provides a maturity score for each of ten security services. The scores range from 0 to 3, where 3 is the best.

The scores are based on how well the product revision meets the security requirements set forth for each security service. These requirements are built from the [Security Best Practices for Non-Voting Election Technology](https://www.cisecurity.org/wp-content/uploads/2019/11/Security-Best-Practices-Non-Voting-Election-Tech-Singles-19-Nov.pdf).

To determine the maturity level for each security service, the RTP responds the following question:

1. Are these technology controls or activities implemented in the product?

There are four permitted answers:

1. No
2. Yes, on some components
3. Yes, on most components or on all components but they are provided manually
4. Yes, on all components that require it

The maturity levels have consistent basic definitions across each security service:

* Maturity level 0: The RTP has not implemented basic best practices across the product
* Maturity level 1: The RTP has implemented the basic level best practices across the product
* Maturity level 2: The RTP has implemented the basic and advanced best practices across the product
* Maturity level 3: The RTP has implemented the basic, advanced, and specialized best practices across the product

## Authentication Capability Maturity

The quality criteria for the Authentication Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Authorization Capability Maturity

The quality criteria for the Authorization Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Boundary Protection Capability Maturity

The quality criteria for the Boundary Protection Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Data Confidentiality Protection Capability Maturity

The quality criteria for the Data Confidentiality Protection Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Data Integrity Protection Capability Maturity

The quality criteria for the Data Integrity Protection Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Injection Prevention Capability Maturity

The quality criteria for the Injection Prevention Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Logging and Alerting Capability Maturity

The quality criteria for the Logging and Alerting Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## Secret Management Capability Maturity

The quality criteria for the Secret Management Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## System Integrity Capability Maturity

The quality criteria for the System Integrity Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

## User Session Management Capability Maturity

The quality criteria for the User Session Management Capability are:

|  |  |
| --- | --- |
| **Maturity Level** | **Quality Criteria** |
| Level 1 | [tk profile level 1] |
| Level 2 | [tk profile level 2] |
| Level 3 | [tk profile level 3] |

# Security Services Architectural Maturity Index

The SSAM index provides scores indicate how well the product’s architecture is built to support each security service. This is a measure of the reliability of the security service and how isolated the security service is from other system changes. These maturity scores are measured during the RABET-V Architecture Review.

The SSCM Index provides a maturity score for each of ten security services. The scores range from 0 to 3, where 3 is the best.

The SSAM Index provides maturity scores across a security service construction stream and a security service use stream. These streams measure maturity levels across different aspects of the system and combine to provide a more complete picture of the security of the product.

## [Security Service Construction](file:///C:/Users/wilso/Downloads/SSAMModel/Security_Service_Construction_Maturity)

|  |  |  |
| --- | --- | --- |
| **Security Service Construction Maturity Levels** | **Quality Criteria** | **Required Activity** |
| * Level 0 |  |  |
| * Level 1: components are well-vetted | Vetted components, ideally by a reputable third party. | * Written in house (partial credit) * Written third-party (full credit) * No vetted services (no credit) |
| * Level 2: components are well-documented | Documentation that the security service is properly used, including illustrative examples of the security service | * Documentation limited in depth or scope (partial credit) * Comprehensively documented (full credit) * No documentation (no credit) |
| * Level 3: a consistent and stable interface integrates with the chosen technology framework | A security service API that is clear, consistent, and straightforward with interfaces that integrate effectively with the chosen technology stack[[1]](#footnote-2) | * Consistent interface but not integrated with available framework extension point (partial credit) * Consistent interface integrated with underlying framework (full credit) * Other (no credit) |

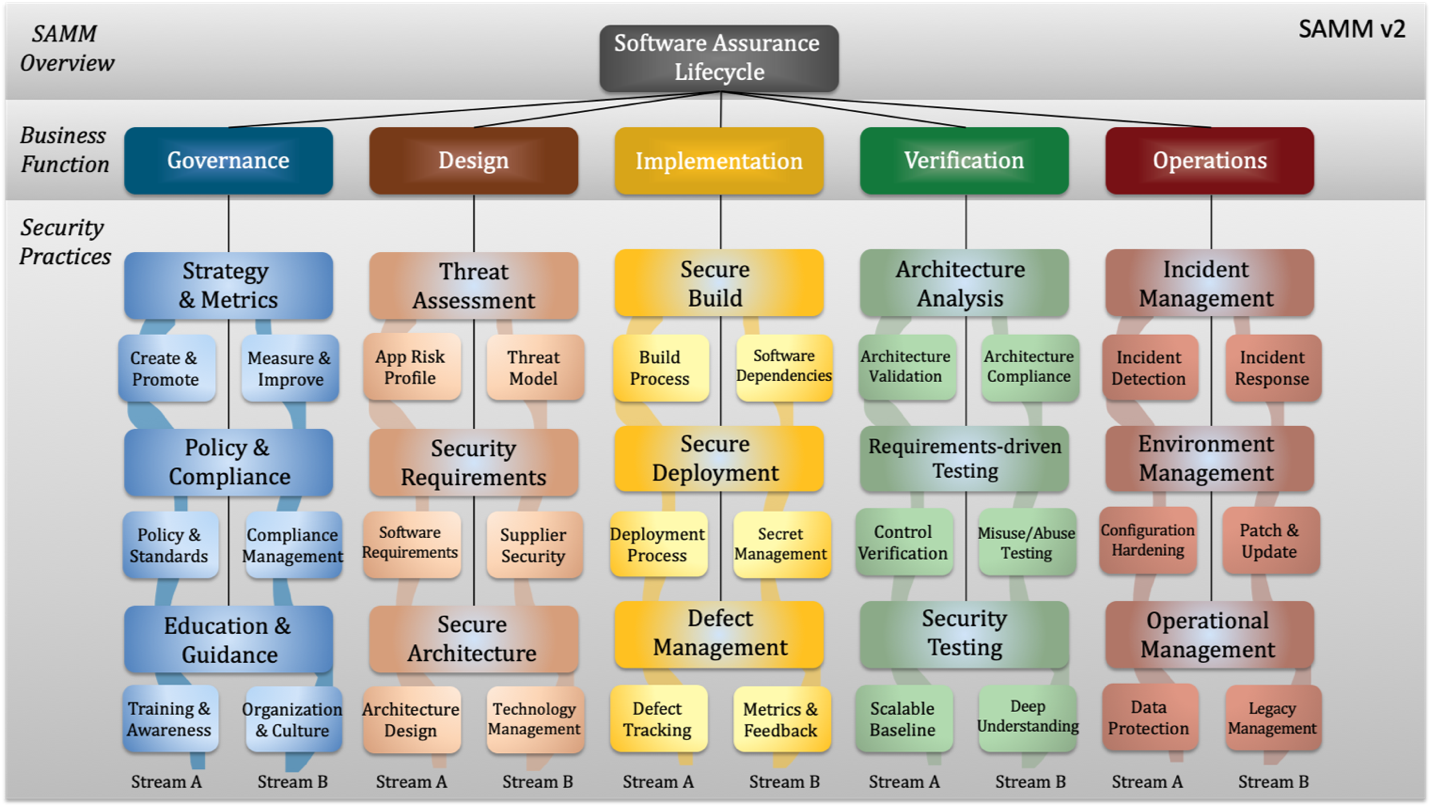
## [Security Service Use](file:///C:/Users/wilso/Downloads/SSAMModel/Security_Service_Usage_Maturity)

|  |  |  |
| --- | --- | --- |
| **Security Service Use Maturity Levels** | **Quality Criteria** | **Required Activity** |
| * Level 0 |  |  |
| * Level 1: dedicated component exists in the system | A minimal number of components provide the service via the same means, ideally one per technical stack or subsystem.[[2]](#footnote-3) | * One component is used for the security service but it is not dedicated to security mitigations (partial credit) * One dedicated component is used to provide the security service within the subsystem (full credit) * Multiple components are involved in providing the security servicer in a subsystem (no credit) |
| * Level 2: security services are centrally configured | The security service is configured through a single component and is used consistently throughout the subsystem[[3]](#footnote-4) | * Mostly configured centrally, with some service details configured at point of use (partial credit) * Configured centrally, or no configuration required (full credit) * Usually or always configured at the point of use (no credit) |
| * Level 3: a consistent and stable interface integrates with the chosen technology framework | Security service components are used throughout the entirety of the product; system defaults enforce use of the security service; user is verifiable[[4]](#footnote-5) | * Consistent use of the security service throughout the application but it’s use is not the default or easily verifiable (partial credit) * Consistent use of the security service which is made easy and verifiable by defaults or reusable components (full credit) * Little to no consistent use of the security service or it’s use is missing from key components (no credit) |

# Software Development Maturity Index

The SDM Index score is measured by the RABET-V Process Review activity and indicates the maturity of the provider’s software development processes for security and usability. The RABET-V SDM score is based on the [OWASP Software Assurance Maturity Model (SAMM)](https://www.owaspsamm.org).

Maturity scores are provided for each of the 17 software development areas (15 SAMM plus Usability and Accessibility). The scores range from 0 to 3, where 3 is the best.



## Accessibility

Accessibility is often overlooked as a development priority. It may be hard for developers without a disability to conceptualize needing or using accessibility features, but it’s easy to find examples that may be possible for anyone to imagine. For example, some software developers developed repetitive stress injuries and turned to speech-to-text aids to continue working in their profession. Beyond the general necessity, adhering to accessibility standards is often a hard requirement for software solutions in many state systems.

|  |  |  |
| --- | --- | --- |
| **Accessibility Maturity Levels** | **Quality Criteria** | **Required Activity** |
| * Level 0 |  |  |
| * Level 1: automated conformance to accessibility guidelines | Performs automated accessibility validation during development. | Use automated testing tools during development for:   * All major releases (partial credit) * All significant changes to user interface functionality (full credit) * Other (no credit) |
| * Level 2: Testing with accessibility tools | Perform accessibility tests with commercial accessibility software and OS-specific features, including using personas and scenarios | Use commercial software, OS-specific features, and personas and scenarios for:   * All major releases (partial credit) * All significant changes to user interface functionality (full credit) * Other (no credit) |
| * Level 3: formal accessibility testing and analysis program | Use of research methods and usability experts to test prototypes with users that have accessibility needs. | Conduct accessibility testing and integrate results for:   * All major releases (partial credit) * All significant changes to user interface functionality (full credit) * Other (no credit) |

## Usability

Usability testing and analysis helps bridge the gap between a solution that meets a set of requirements and a solution that meets the needs of the organization, people, and processes. Meeting usability objectives is the distinction is between a solution that people want to use—i.e. meets a set of requirements and usability needs—versus one they don’t—i.e. solely meets a set of requirements.

Users will attempt to reduce friction in completing their desired task. A poorly designed user experience will result in users finding workarounds, often circumventing well-intentioned security controls. For a product to achieve the risk mitigation intended by the security requirements, it must integrate usability principles with security controls and, thus, an organization's maturity in implementing usability is critical to its security outcomes.

|  |  |  |
| --- | --- | --- |
| **Usability Maturity Levels** | **Quality Criteria** | **Required Activity** |
| * Level 0 |  |  |
| * Level 1: formally established feedback loops with customers | Established processes for receiving feedback from customers and incorporating that feedback into the product | Incorporation of feedback into products for:   * All major releases (partial credit) * All updates involving user-facing functionality (full credit) * Other(no credit) |
| * Level 2: deploy enhanced feedback capabilities | Interview users, accept feedback directly through the product, collect logs and analytics through the product, or other similar approaches; from these, product form reports on findings and plans for incorporating feedback | Use commercial software, OS-specific features, and personas and scenarios for:   * Most major releases (partial credit) * All significant changes to user interface functionality (full credit) * Other (no credit) |
| * Level 3: formal accessibility testing and analysis program | Formal research on the business processes and users’ behaviors, and conduct usability studies with users interacting with a prototype or version of the software solution. | Conduct formal usability testing and integrate results for:   * Most major releases (partial credit) * All significant changes to user interface functionality (full credit) * Other (no credit) |

# Documentation Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Description** | **Produced By** | **Disclosure(s)** |
| Registered Technology Provider Request (excluding Organizational Security Audit) | Information submitted by RTP to the Administrator | Registered Technology Provider | Public |
| Organizational Security Audit | Information submitted by RTP to the Administrator annually | Registered Technology Provider | Subscribers |
| Subscriber Agreement | The agreement completed by the Subscriber to be given access to sensitive information. | Subscriber | List of subscribers will be maintained on the RABET-V Portal |
| Product Goals | Overview of what the product is intended to do | Registered Technology Provider | Public |
| Expected Usage | Statements of how the product is intended to be used | Registered Technology Provider | Public |
| Security Claims | Claims of which requirements are met by the product | Registered Technology Provider | Subscribers |
| Process Descriptions | Descriptions of how the provider does product development | Registered Technology Provider | RABET-V Administrator only |
| Architecture Documentation and Diagrams | Documentation on how the product is constructed | Registered Technology Provider | RABET-V Administrator only |
| Third-Party Component Details | Listing of the 3rd party software packages used by or included in the product | Registered Technology Provider | RABET-V Administrator only |
| User Documentation | Documentation intended to help non-technical users use the product | Registered Technology Provider | Subscribers |
| Product Revision Submission Artifacts | Outputs of the providers internal product development process | Registered Technology Provider | RABET-V Administrator only |
| Submission Review Checklist | Checklist completed during the Submission Review activity | RABET-V Administrator, or agent | Registered Technology Provider |
| SAMM Toolkit and interview session notes | Interview and scoring toolkit used for the Process Assessment | RABET-V Administrator, or agent | Registered Technology Provider |
| Reliable Artifacts Evaluation | Produced by the Process Assessment and used for Testing Rules | RABET-V Administrator, or agent | Registered Technology Provider |
| Security Service Architecture | Produced by the Architecture Review and used for maturity evaluation | RABET-V Administrator, or agent | Registered Technology Provider |
| List of Applicable and Not Applicable Security Requirements | A determination made during the Security Claims Review based on the product goals and expected usage | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| Security Service Architectural Maturity Scores | Scores developed during the Architecture Review activity | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| Software Development Maturity Scores | Scores developed during the Process Assessment activity | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| Testing Rules Decision Tree | Decision formula for determining how to test product changes | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| Product Verification Test Results | Results produced by testing the product against the requirements | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| Security Service Capability Maturity Scores | Scores developed based on the results of the Product Verification activity | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| RABET-V Product Provider Report | Report which captures the main outputs of the RABET-V iteration | RABET-V Administrator, or agent | Registered Technology Provider, Subscribers |
| RABET-V Product Public Report | Report which captures the main outputs of the RABET-V iteration | RABET-V Administrator, or agent | Public |

# RABET-V Glossary

### 1st Degree Component

a component that provides or configures one of the 10 [security services](#security-service). Components are determined to be 1st or [2nd degree components](#2nd-Degree-Component) in the Architecture Review.

### 2nd Degree Component

a component that uses one of the components which provide or configure a [security service](#security-service). Components are determined to be 1st or 2nd degree components in the Architecture Review.

### activity

a self-contained aspect of the RABET-V Program. Each activity has a process with inputs, outputs, and a workflow.

### data criticality label

a label indicating the sensitivity of the data the component is handling. This may be thought of as a label of “integrity”. This is measured by the impact of the data being manipulated to an unknown or incorrect value. Criticality can be determined by examining a component’s exposed interfaces.

### data sensitivity label

a label indicating the sensitivity of the data the component is handling. This may be thought of as a label of “confidentiality”. This is measured by the impact of the data being exposed to an unauthorized party. Sensitivity can be determined by examining a component’s exposed interfaces.

### functions

a discrete piece of functionality provided by the [product](#product). Represented as a “[process](#process)” in Microsoft’s [Threat](#threat) Modeling Tool, and a “[port](#port)” in the UML Component diagram.

### port

a bundle of interfaces that provides system functionality.

### product

an election technology submitted to RABET-V.

### product revision

a specific version of the [product](#product) submitted to RABET-V.

### RABET-V Administrator

the organization responsible for overseeing and executing the RABET-V Program. CIS is the administrator for the pilot program.

### RABET-V Iteration

a complete cycle through the RABET-V activities with a unique [product revision](#product-revision). The first iteration is called the Initial Iteration.

### RABET-V Subscriber

a state or local jurisdiction who has requested access to sensitive RABET-V reporting

### Registered Technology Provider

an organization that develops election technology and has met the minimum requirements to become a RABET-V Registered Technology Provider.

### required security services

mechanisms used to provide confidentiality, integrity authentication, source authentication and/or support non-repudiation of information.

### security service

a capability that supports one, or many, of the security goals (NIST definition). Examples of security services are key management, access control, and authentication. RABET-V defines 10 security services which are used to create the [Security Service Capability Maturity (SSCM)](#Security-Service-Capability-Maturity-(SSCM)) scores and the [Security Services Architectural Maturity (SSAM)](#Security-Services-Architectural-Maturity-(SSAM)) scores.

### Security Service Capability Maturity (SSCM)

a set of maturity scores for each of the ten [security services](#security-service) that is one of the primary metrics reported by RABET-V.

### security service catalog

a set of [security services](#security-service) identified by RABET-V to mitigate [threats](#threat).

### security service label

mechanisms used to provide confidentiality, integrity authentication, source authentication and/or support non-repudiation of information.

### Security Services Architectural Maturity (SSAM)

a maturity score created by the RABET-V Architecture Review activity to indicate how well the [product](#product)’s architecture is defined to provide the [security services](#security-service).

### Security Services Architecture

an architectural view created in the Architecture Review which identifies components and maps them to the 10 [security services](#security-service).

### services

a system level component that provides data processing capabilities.

### Software Development Maturity (SDM)

a maturity score measured by the RABET-V [Process](#process) Review activity to indicate maturity of the provider’s software assurance processes. The RABET-V SDM score is based on the OWASP Software Assurance Maturity Model (SAMM).

### testing rules

a set of rules specific to the technology provider and [product](#product) which determine how changes to that product will be verified during [RABET-V iterations](#RABET-V-Iteration).

### threat

a role of a situation that my lead to one ore more related incidents or failures.

1. To protect against changes in the underlying implementation of the security service, it should expose stable interfaces. Ideally, the security service should integrate with the underlying framework used (if any). For example, a Java application implementing Authentication could use the platform’s built in LoginModule. Platforms that do not provide extension points for a given security service will not be penalized. [↑](#footnote-ref-2)
2. This should not be construed to discourage defense-in-depth approaches. Two components that provide different mitigations to the same threat are acceptable. For example, a security service component may provide input sanitization to defend against SQL injection. Another may provide query parameterization to defend against the same. [↑](#footnote-ref-3)
3. Ideally, a single component is responsible for the configuration of the security service. This criteria rewards approaches that centralize the configuration of security services. If the security service does not support central configuration, and no wrapper has been written, no credit will be given. [↑](#footnote-ref-4)
4. This item measures how widespread and appropriate the security service components are used in the application. Ideally, the system is setup to use the security service throughout the application and it does not have to be invoked many times by different components. Where it does need to be invoked, there are reusable components (i.e. base classes, base page, etc.) which invoke it by default. [↑](#footnote-ref-5)