RABET-V Working Model

Version 0.1.0

April 23, 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 RABET-V Pilot Program 4](#_Toc38531095)

[1.1 Steering Committee 4](#_Toc38531096)

[1.2 Pilot Participants 4](#_Toc38531097)

[1.3 Technology Advisory Committee 5](#_Toc38531098)

[1.4 Program Administration and Research 5](#_Toc38531099)

[1.5 Pilot Process 5](#_Toc38531100)

[2 Pilot Research Questions 6](#_Toc38531101)

[2.1 Time and Cost Implications 6](#_Toc38531102)

[2.2 Market Maturity Implications 6](#_Toc38531103)

[2.3 Pre-Review Assessment Implications 6](#_Toc38531104)

[2.4 Technical Evaluation Implications 7](#_Toc38531105)

[3 RABET-V Working Model 8](#_Toc38531106)

[3.1 Introduction 8](#_Toc38531107)

[3.2 Guiding Framework 8](#_Toc38531108)

[3.3 Registered Technology Providers 9](#_Toc38531109)

[3.4 Provider Deregistration and Product Delisting 9](#_Toc38531110)

[3.5 RABET-V Public Portal 10](#_Toc38531111)

[3.6 Maturity Indexes 10](#_Toc38531112)

[3.7 Process Activity Descriptions 10](#_Toc38531113)

[4 RABET-V Security Services 11](#_Toc38531114)

[4.1 Security Service Definitions 11](#_Toc38531115)

[5 RABET-V Maturity Indexes 13](#_Toc38531116)

[5.1 RABET-V Security Services Capability Maturity (SSCM) Index 13](#_Toc38531117)

[5.2 RABET-V Security Services Architectural Maturity (SSAM) Index 13](#_Toc38531118)

[5.3 RABET-V Software Development Maturity (SDM) Index 14](#_Toc38531119)

[6 RABET-V Activity Process Details 15](#_Toc38531120)

[6.1 Provider Submission 15](#_Toc38531121)

[6.2 Submission Review Process 18](#_Toc38531122)

[6.3 Process Assessment Methodology 19](#_Toc38531123)

[6.4 Architecture Review Methodology 27](#_Toc38531124)

[6.5 Security Claims Validation 33](#_Toc38531125)

[6.6 Testing Rules Determination 37](#_Toc38531126)

[6.7 Product Verification Activity 40](#_Toc38531127)

[6.8 Reporting Process 43](#_Toc38531128)

[7 SSCM Model 46](#_Toc38531129)

[7.1 Authentication Capability Maturity 46](#_Toc38531130)

[7.2 Authorization Capability Maturity 47](#_Toc38531131)

[7.3 Boundary Protection Capability Maturity 48](#_Toc38531132)

[7.4 Data Confidentiality Protection Capability Maturity 49](#_Toc38531133)

[7.5 Data Integrity Protection Capability Maturity 50](#_Toc38531134)

[7.6 Injection Prevention Capability Maturity 52](#_Toc38531135)

[7.7 Logging and Alerting Capability Maturity 53](#_Toc38531136)

[7.8 Secret Management Capability Maturity 54](#_Toc38531137)

[7.9 System Integrity Capability Maturity 55](#_Toc38531138)

[7.10 User Session Management Capability Maturity 56](#_Toc38531139)

[8 SSAM Model 58](#_Toc38531140)

[8.1 Security Service Construction Maturity 58](#_Toc38531141)

[8.2 Security Service Usage Maturity 59](#_Toc38531142)

[9 SDM Model 62](#_Toc38531143)

[9.1 Strategy & Metrics 62](#_Toc38531144)

[9.2 Policy & Compliance 62](#_Toc38531145)

[9.3 Education & Guidance 62](#_Toc38531146)

[9.4 Threat Assessment 62](#_Toc38531147)

[9.5 Security Requirements 62](#_Toc38531148)

[9.6 Secure Architecture 62](#_Toc38531149)

[9.7 Secure Build 62](#_Toc38531150)

[9.8 Secure Deployment 62](#_Toc38531151)

[9.9 Defect Management 62](#_Toc38531152)

[9.10 Architecture Assessment 63](#_Toc38531153)

[9.11 Requirements Testing 63](#_Toc38531154)

[9.12 Security Testing 63](#_Toc38531155)

[9.13 Incident Management 63](#_Toc38531156)

[9.14 Environment Management 63](#_Toc38531157)

[9.15 Operational Management 63](#_Toc38531158)

[9.16 Accessibility 63](#_Toc38531159)

[9.17 Usability 65](#_Toc38531160)

[10 RABET-V Glossary 67](#_Toc38531161)

# RABET-V Pilot Program

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

In this repository, you will find three main areas:

* [Working Model](file:///C:\Users\wilso\Downloads\WorkingModel\). The Working Model defines the RABET-V process details. Read through the working model to get an idea of how RABET-V works in detail.
* [Research Plan](file:///C:\Users\wilso\Downloads\ResearchPlan\). The Research Plan discusses how we intend on evaluating RABET-V throughout the Pilot Program. Read this to understand the pilot project.
* [Economic Model](file:///C:\Users\wilso\Downloads\EconomicModel\). The Economic Model proposes various methods of deploying and operating a RABET-V process in the United States for non-voting election technology verification.

## 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

## 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

## 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 Software Solutions Division
* Gema Howell, NIST
* Mary Brady, NIST
* Gordon Gillerman, NIST
* Lisa Carnahan, NIST
* 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 Program will first establish a detailed version of the RABET-V process called the [RABET-V Working Model](file:///C:\Users\wilso\Downloads\WorkingModel). This version will detail how each activity will be conducted. The Working Model will be iteratively reviewed by the program Steering Committee and modified as necessary.

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

The RABET-V process will be conducted on real products from Pilot Program participants 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 Program will evaluate the value of the activities, along with the time and cost, and conclude with recommendations on the best approach.

The Pilot Program 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 Working Model

RABET-V is a flexible, risk-based, and cost-effective election system verification process that will expedite verification of election systems while providing assurances of security. reliability, and usability. 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. This Working Model describes RABET-V Process in detail. Along with this version of the Working Model, CIS is issuing two companion documents: (1) a Research Plan that describes how CIS intends to iterate and confirm that the framework proposed in this Working Model provides assurances of security, reliability, and usability for non-voting election technology products and (2) the Economic Model for managing RABET-V post pilot.

## Introduction

### Program Goal

The goal is the program is to develop a rapid, reliable, and cost-effective system for verifying the security of non-voting election systems. Verification will provide adequate assurances of security, reliability, usability. Assurances are evidence-based grounds for confidence that the system’s features are effective in their application. The RABET-V Program is intended to build stakeholder confidence that participating products and organizations have demonstrated capabilities to build, test, monitor, and maintain the reliability, usability, and security of their election technology solution.

### 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. There are voting systems and non-voting systems. 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 during the 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)

## Guiding Framework

This Working Model is one approach to meeting the process described by the Center for Internet Security (CIS) in the [How to Improve Election Technology Verification White Paper](file:///C:\Users\wilso\Elections_Tech-Ver-White_Paper-2020-0121.pdf).

## Registered Technology Providers

The RABET-V Program is a continuous operation with participating technology providers known as Registered Technology Providers (RTPs). To be a RABET-V Registered Technology Provider, the technology provider must submit a complete RTP request, have at least one product verified by RABET-V, and agree to the RABET-V Program Commitment. All current Registered Technology Providers will be listed on the RABET-V Program Portal. For the Pilot Program, each pilot participant will be considered an RTP.

### 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
* Point of Contact
* Website URL
* Company Description

### Program Commitment

Technology providers who wish to remain RABET-V registered technology providers must agree to the RABET-V Program Commitment. The commitment establishes the ethical and responsible behavior expected by all program providers. The program commitment is subject to change. The Commitment includes:

* Accurate representation of the product capabilities and its security provisions to RABET-V reviewers, customers, and other stakeholders
* Organization implementation and regular assessment of an organizational security framework like the CIS Controls. Organization security audits must be performed regularly and the report provided to RABET-V. 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 Registered Technology Provider and/or delisting of the provider’s products.

* Inaccurate representation - if the vendor is found to have intentionally mislead RABET-V authorities or its customers as to the organization or the product’s capabilities, the vendor risks being deregistered from the program and all product’s delisted from the RABET-V registry.
* Lacking organization security - if the vendors fails to subscribe to an organizational security framework, like the CIS controls, and maintain regular audits, the vendor risks being deregistered from the program and all product’s delisted from the RABET-V registry.
* Lacking product maintenance - if the vendor is no longer properly supporting a product with regular monitoring and maintenance, RABET-V will delist the product from its registry.

## RABET-V Public Portal

### Purposes

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

* Lists all Registered Technology Providers (RTP)
* 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 (if one completed)

### RABET-V Subscriber Access vs. Public

Many of RABET-V reports are made public, but some sensitive reports are not made publicly available. They are, however, available to RABET-V Subscribers. A RABET-V Subscriber is a Federal, State, or local election authority that has requested access to all RABET-V reports and has agreed to treat the information as sensitive—i.e. not to be shared with the public or other organizations.

## Maturity Indexes

RABET-V evaluates the security, reliability, and usability of products and provides a set of maturity scores for each product version. RABET-V provides three maturity indexes:

* [Security Service Capability Maturity (SSCM) Index](file:///C:\Users\wilso\Downloads\MaturityIndexes\Security_Services_Capability_Maturity_Index) - identifies the product’s current capability level measured across ten security services.
* [Software Development Maturity (SDM) Index](file:///C:\Users\wilso\Downloads\MaturityIndexes\Software_Development_Maturity_Index) - identifies the maturity of the technology provider’s software development processes measured across 15 security practices and 2 usability/accessibility practices.
* [Security Services Architectural Maturity (SSAM) Index](file:///C:\Users\wilso\Downloads\MaturityIndexes\Security_Services_Architectural_Maturity_Index) - identifies the maturity and reliability of the product architecture to support each of the ten security services.

## Activity Process Descriptions

The RABET-V Process consist of eight activities, beginning with Product Submission. The Activity Process Descriptions are provided in Section 6.

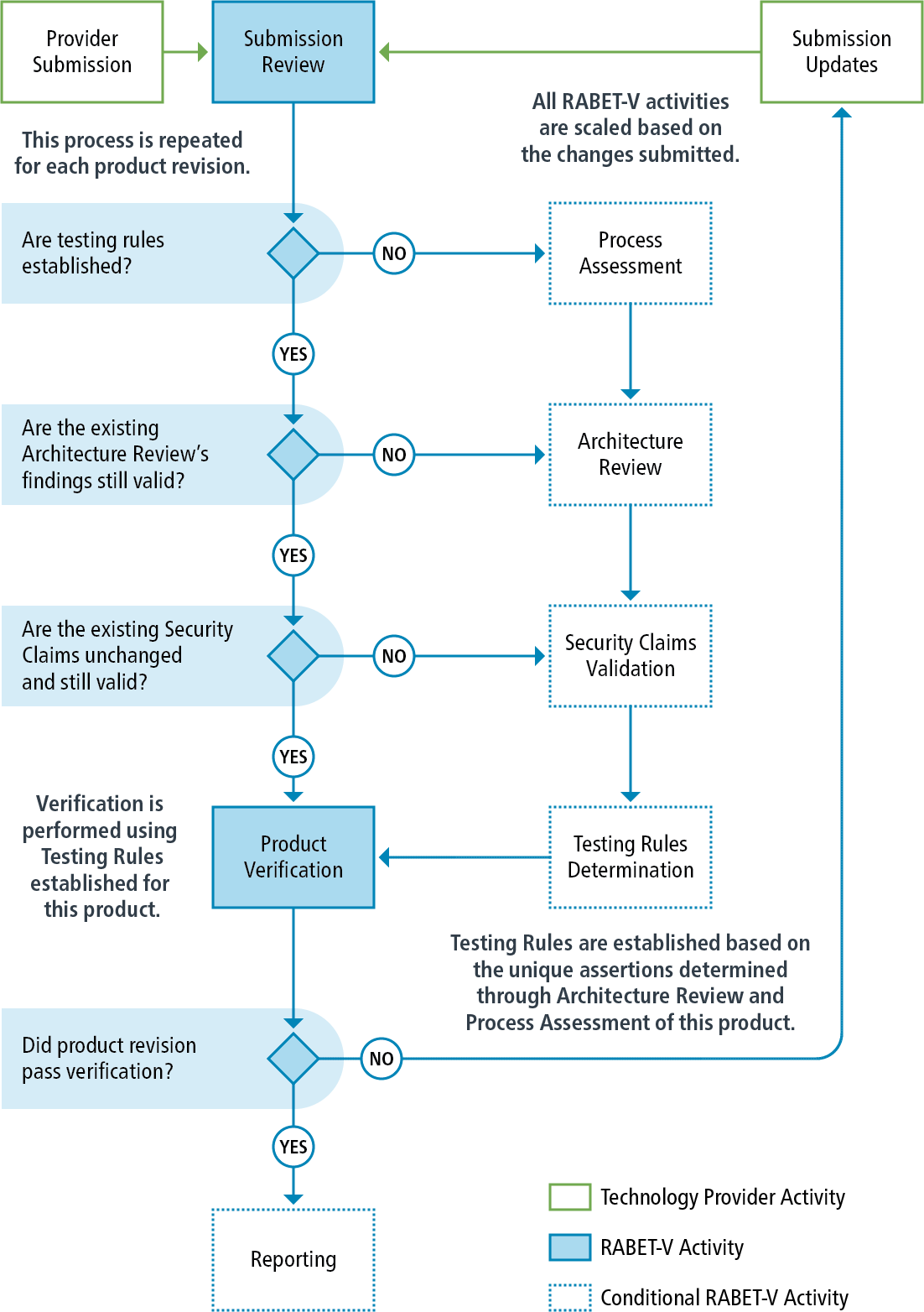


Figure - RABET-V Workflow

# RABET-V Security Services

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

## Security Service Definitions

### 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]

### Authorization

The right or a permission that is granted to a system entity to access a system resource. [NIST SP 800-82 Rev. 2]

### Injection Prevention

The rejection or sanitization of data input and output to ensure malicious executable code is not executed.

### 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]

### User Session Management

A persistent interaction between a subscriber and an end point. [Adapted from NIST SP 1800-17b]

### Logging/Alerting

The systemic management and monitoring of the events occurring within an organization’s systems and networks. [Adapted from NIST SP 800-92]

### 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]

### 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]

### 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]

### 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

RABET-V evaluates the security, reliability, and usability of products and provides a set of maturity scores for each product version. RABET-V provides three maturity indexes:

* [Security Service Capability Maturity (SSCM) Index](file:///C:\Users\wilso\Downloads\Security_Services_Capability_Maturity_Index) - identifies the product’s current capability level measured across ten security services.
* [Software Development Maturity (SDM) Index](file:///C:\Users\wilso\Downloads\Software_Development_Maturity_Index) - identifies the maturity of the technology provider’s software development processes measured across 15 security practices and 2 usability/accessibility practices.
* [Security Services Architectural Maturity (SSAM) Index](file:///C:\Users\wilso\Downloads\Security_Services_Architectural_Maturity_Index) - identifies the maturity and reliability of the product architecture to support each of the ten security services.

## RABET-V Security Services Capability Maturity (SSCM) Index

The SSCM Index provides a set of maturity scores for each of the ten [security services](file:///C:\Users\wilso\Downloads\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).

### Maturity Score Ranges

Maturity scores are provided for each of the 10 security services. A green, yellow, and red range is provided for each area to indicate optimal, acceptable, and unacceptable score ranges.

The range values will be set by the Verification Authority and may be overwritten by individual states who wish to establish their own ranges of optimal, acceptable, and unacceptable.

There are no ranges established for the pilot program.

Green: TBD

Yellow: TBD

**Red:** TBD

### SCCM Model

The SCCM Model is detailed in Section 7.

## RABET-V Security Services Architectural Maturity (SSAM) Index

The SSAM index provides scores which indicate how well the product’s architecture is built to provide the 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. See [Architecture Review Methodology](file:///C:\Users\wilso\Activities\Architecture_Review_Methodology) for more information on this maturity index.

### Maturity Score Ranges

Maturity scores are provided for each of the 10 security services. A green, yellow, and red range is provided for each area to indicate optimal, acceptable, and unacceptable score ranges.

The range values will be set by the Verification Authority and may be overwritten by individual states who wish to establish their own ranges of optimal, acceptable, and unacceptable.

There are no ranges established for the pilot program.

Green: TBD

Yellow: TBD

**Red:** TBD

### SSAM Maturity Model

The SSAM Maturity Model is detailed in Section 8.

## RABET-V Software Development Maturity (SDM) Index

The Software Development Maturity 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 Score Ranges

Maturity scores are provided for each of the 17 software development areas (15 SAMM plus Usability and Accessibility). A green, yellow, and red range is provided for each area to indicate optimal, acceptable, and unacceptable score ranges.

The range values will be set by the Verification Authority and may be overwritten by individual states who wish to establish their own ranges of optimal, acceptable, and unacceptable.

There are no ranges established for the pilot program.

Green: TBD

Yellow: TBD

**Red:** TBD

# RABET-V Activity Process Description

The Rapid Architecture-Based Election Technology Verification (RABET-V) process consists of eight total activities, five of which are conditional activities that are scaled to meet the needs of the particular review. This scaling provides a rapid, risk-based testing strategy informed by the product’s architecture, the developer’s processes, and their security claims.

## Provider Submission

The RABET-V process begins with a product submission.

### 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 Registered Technology Provider (RTP) that will be used throughout the RABET-V process.

An Initial Submission requires all items listed in the remainder of the Provider Submission Process.

#### 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. It can improve the likelihood of a smooth process by engaging the [Verification Authority](file:///C:\Users\wilso\RABET-V_Glossary) 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 items listed in the remainder of the Provider Submission Process.

### Submission Items

#### Product Goals – Initial Submission Only

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.

This description will be used by the [Verification Authority](file:///C:\Users\wilso\RABET-V_Glossary) in the Submission Review activity to determine if the stated security claims align with with the product goals. For example, if the Product Goals include managing sensitive voter information, the Verification Authority 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 Only

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.

This description will be used by the [Verification Authority](file:///C:\Users\wilso\RABET-V_Glossary) in the Submission 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 Only

The Product Security Claims statement is a listing of security requirements met by the product. While the provider may use any set of security requirements, there is a strong preference for leveraging widely-accepted and publicly available control sets to the maximum extent possible. For instance, the provider may have specific security requirements for the product, but should map those requirements to the most recent revisions of the CIS Controls or NIST SP 800-53.

For each implemented requirement, the provider will describe how they implement it. 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 for each product to implement a unique approach to the security of their application that is specific to its goals and usage.

#### Process Descriptions – Initial Submission Only

The Process Descriptions statement is about the provider’s development and operating environment and organizational maturity and controls. There are two parts the statement: the software development maturity and the organizational security maturity.

For the software development maturity portion of the Process Description statement, the provider should use OWASP Software Assurance Maturity Model (SAMM) as the basis for its evaluation. SAMM will also provide the basis for the interview portion of the process assessment.

For the organizational security maturity of the Process Description statements, the provider should use the most recent version of the CIS Controls. The maturity of the provider’s organizational security approach will be determined based on its adherence to each of the CIS Controls Implementation Groups.

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 Only

The most up to date documentation is requested. Architecture can be described textually, using diagrams, or a combination of the two. RABET-V expects documentation regarding 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 we would prefer if the provider followed our examples, which are based on UML Component Diagrams.

The 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 Only

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 which are not internally developed, each unique version of the following will be considered individual components of the system:

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

RABET-V Architecture Review will not further deconstruct these elements of the solution. The assumption of the above rule is that any change to 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, 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.

The 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.

#### Revision Submission Artifacts

A provider can submit a product revision to RABET-V at any time. Engaging the [Verification Authority](file:///C:\Users\wilso\RABET-V_Glossary) about upcoming changes and consulting the existing Testing Rules (if exist) will help a technology provider better prepare their submission.

All Revision Submissions require the following artifacts:

* Change list - Indicates which components have changed and what level of change was made. It should reference the components identified in the architecture review.
* 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
* Desired Deployment Date - Target date for deploying the product revision in a production environment.
* 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 Verification Authority.

### Submission

Once the Initial Submission or Revision Submission package is complete, it should be submitted electronically to the RABET-V certifying authority (TBD). For the pilot, this information will be submitted to a secure file upload portal (TBD).

## Submission Review Process

Once the provider has made their submission, the RABET-V Verification Authority must review the submitted information and determine which RABET-V activities are necessary.

### 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 for this submission

### Workflow

#### 1. Review package for completion.

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

##### 1a. 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.

##### 1b. 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.

#### 2. Validate change list

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

* Reliable: skip or streamline change list validation can be skipped or streamlined [tk what does streamlined mean?]
* Otherwise: validate the change list by manual or automated means

Record the result in the Submission Review Checklist.

#### 3. Determine if Process Assessment activity is necessary

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

* The submission is an Initial Submission
* 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
* It has been more than 18 months since the last Process Assessment was performed
* Artifacts provided by provider indicate a significant process change has occurred with provider

Record the result in the Submission Review Checklist.

#### 4. Determine if Architecture Review activity is necessary

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

* The submission is an Initial Submission
* 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
* The change list indicates the addition, removal, or modification of architectural components since the last Architecture Review

Record the result in the Submission Review Checklist.

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

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

* The submission is an Initial Submission
* 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
* The change list indicates that a 1st degree architectural component has been modified
* The change list indicates that prior Security Claims Validation findings need to be reviewed

Record the result in the Submission Review Checklist.

## Process Assessment Methodology

The RABET-V Process Assessment Methodology 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 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.

RABET-V adds one area to SAMM to cover usability and accessibility. This becomes the 6th area for the [Software Development Maturity (SDM)](file:///C:\Users\wilso\MaturityIndexes\Software_Development_Maturity_Index) scores.

### Inputs

* Process descriptions
* Interviews with technology provider

### Outcomes

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

### Workflow

#### 1. Review Existing Documentation

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

* Process related documents that help define which processes are followed related to software activities at the organization.
* Artifacts from completed activities related to the above policy and compliance or process related activities. Just a representative sample is needed.

#### 2. Discussion Sessions

These sessions are for interactive discussions with the different roles supporting the efforts related to software development at an organization. The interviews will normally involve two interviewers and each will last approximately 60-90 minutes in length. 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 role that would be interviewed, however representatives from the logical business units are also useful candidates for interviewing:

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

#### 3. 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 submit a request for 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. Manually generated change lists created by personnel outside of the software development process produced inaccurate and incomplete change lists. However, automated change lists built from the central source code repository and reviewed by system architects and product owners yield more accurate and reliable change lists. 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 one themselves. If the change list is not reliable, the RABET-V process will explore other ways to produce an accurate change list - which may take addition time and resources.

##### Automated Configuration Assessments

Security configurations are a major part of ensuring that systems are implementing proper 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 3rd party components for known vulnerable versions of software. For technology providers who are found to be regularly performing automated vulnerability scans on the product networks and software, their results can be 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 which are robust and accurate. For technology providers who 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 technology provider’s commitment to maintaining their test suites.

##### 3rd Party Security Analysis

It is highly encouraged for technology providers to receive regular, in-depth security audits on their systems. For example, there are audits which 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.

#### 4. 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:

* High level executive summary of the process, findings, maturity level score and tailored recommendations in a PowerPoint format
* Completed SAMM Toolbox in Excel format
* Interview session notes in a Word document format

### Maturity Levels

The SAMM assessment provides a maturity score for each security practice. These maturity scores provide assistance determining risk of future changes.

In this section, we describe the implication of maturity scores on the RABET-V testing rules and verification.

#### Governance

##### Strategy & Metrics

At a base level, the organization should identify the means of measuring the effectiveness of any security program. These indicators may be different based on the organization, but could include Mean Time to Remediation, Flaw Creation Rate, and Portion of Application covered by Automated Security Testing. Level two involves developing a strategy to ensure application security and setting goals for each of the indicators to monitor and track progress.

##### Policy & Compliance

Level one of Policy & Compliance expects that the organization has a common set of policies and standards that govern all aspects of software development. The policies protect the integrity of its computing environment, safety and privacy of the data, and maturity of the software development life-cycles. The standards set requirements for technologies—such as languages and frameworks—used within the organization.

The base level of Compliance Management involves creating a comprehensive list of all compliance requirements and what triggers potentially put an application in scope. One useful artifact is a compliance matrix—aimed at the organizational level rather than the application level—which provides a basic understanding of useful compliance requirements. At level two, each external compliance obligation should be mapped to a well-defined set of application requirements and the organization has procedures in place to verify compliance against the requirements.

##### Education & Guidance

Level one of Education & Guidance requires the creation of a training program that involves various different topics, such as specific application best practices that is incorporated as part of the onboarding process. Higher levels of this practice involves the additional coverage of the training program to other members of the workforce and deeper coverage of the topics associated with SDLC. Having a general education program in place that details who the program applies to and the topics covered by the program could be considered as a starting point.

#### Design

##### Threat Assessment

The practices in this stream focus upon building and maintaining the core risk profiles of the organization and assuring that they’re aligned to both the business needs of the organizations and risk the organization faces. Having both the ability to access the final risk registry and threat model methodology used would be beneficial for the assessment and could present a reasonable starting point.

##### Security Requirements

At a base level the organization should have a documented set of security requirements, with higher maturity levels having it in a more standardized notation and including active participation and development from the application development team. Having a well documented set of security requirements should be considered a baseline for verification efforts.

##### Security Architecture

The second level of maturity dictates that organizations should have an established list of recommended technologies and a list of resusable security services that are use to support each of the products. These two could be considered useful artifacts for more advanced verification efforts.

#### Implementation

##### Secure Build

At a base level, the organization should have a record of all dependencies used in the production environment. The second level of maturity dictates that this list of dependencies should be assessed as to whether each dependency meets a set of criteria (e.g. no known vulnerabilities, up-to-date version, actively supported and maintained). The third level of maturity is reached with the development of a formalized whitelist of approved dependencies and versions, which is integrated into the build process.

##### Secure Deployment

In the base of the secure deployment maturity organizations are required to have a documented process for conducting deployments, including the appropriate use and handling of application secrets, with the higher levels of maturity requiring the use of automated processes that leverage security checks as part of the process.

##### Defect Management

With level one of Defect Management, an organization should establish a single location for tracking defects in an application. The defects should also be clearly defined and may include threat assessments, penetration tests, and output from static and dynamic analysis scanning tools. The organization should also have a rudimentary defect classification system in place to prioritize the handling of defects based on risk. At level two, a single severity scheme is in place across the organization and each severity level has clearly defined service-level agreements (SLAs) for resolution.

#### Verification

##### Architecture Assessment

The Architecture Assessment builds on the work from Governance and Design to identify compliance issues and develop mitigations for known threats. At a base level, the architecture is reviewed for structural stability and all missing security controls are logged as defects. Higher levels of maturity will also include activities periodic re-evaluations of the threat assessments as well as the review of the internal and external requirements, which are either logged as defects if unaddressed for organizations of the highest level of maturity.

##### Requirements-driven Testing

Level one of Requirements-driven Testing requires testing the correct functioning of the standard security controls—i.e. aspects of the service that control confidentiality, integrity, and availability. Testing at this level should also cover fuzz testing the main input parameters and the inspection of as many application crashes as possible for security impact. In addition for organizations for maturity level 2 or higher will have security tests automated and documented using a standardized framework or DSL. For organizations with the highest level of maturity or organizations that have a high availability requirement, should also conduct security stress testing to assure that the application can perform in adverse circumstances.

##### Security Testing

Security Testing involves both in-depth testing based on knowledge of an application and its business logic and automated testing for scalability. For organization’s of all maturity this testing should be done in an automated fashion with some level of manual review for high risk components. Organizations with a higher level of maturity will have this security testing build into their build and deploy process with a feedback loop that helps improve the organization’s secure development practices.

#### Operations

##### Incident Management

At a base level, the expectation is for the organization to perform best-effort incident detection. This would involve designating a contact point and person for the creation and handling of security events, developing an incident documentation process, analyzing log data based on the log retention schedule, and matching the frequency of analysis to the level of importance of the application.

##### Environment Management

Eventually, every aspect of the application or service starts to show its age. For this reason, over time, parts of the application will need to be updated, patched, or culled. At a base level, this process is performed on a best-effort basis using available information for configuration hardening and attempting/testing updates to system and application components. At the second—and more preferable—level, this process is performed using formal processes and baselines which should be well-documented, timely, and completed according to a schedule. The final level of maturity has the organization assessing the technology against the established baseline with a methodology in place both to review the baseline and remediate any noncompliant systems.

##### Operational Management

The Operational Management of applications within an organization is based around the people and processes. At a base level, basic data protection practices—e.g. knowing what’s processed and stored in an application, what is the sensitivity level of the data, and how to prevent compromising the data—should be in place. Unused application should be identified and decommissioned as often as possible. There should also be a process in place for sunsetting legacy applications and prevent unexpected disruptions for customers and users.

### Other Possible Reliable Artifacts

One of the advantages of having higher maturity scores with SAMM is that the provider’s processes produce more reliable artifacts which can be used by RABET-V. These artifacts may reduce the assessed risk and/or eliminate the need for the RABET-V process to reproduce these artifacts. The evaluation will determine which of these artifacts are reliable and robust enough to be used by RABET-V.

The following are the artifacts required by certain SAMM maturity levels which may be useful in a RABET-V iteration.

#### Governance/Measure and Improve Stream

Maturity Level 2 and 3 have defined KPIs reflecting application security. The KPIs and the current KPI results can be used by the RABET-V process to assess the security health of the system. For example, one KPI may be the number of identified vulnerabilities and how quickly they were addressed. This information can assist RABET-V evaluators to determine the effectiveness of current security controls and response mechanisms. In addition SAMM requires the maintenance and documentation of an organization’s risk management process, one such artifact that may be of value to the RABET-V evaluators is an updated risk profile that details the outputs of the organization’s risk assessment.

#### Policy and Compliance/Policy and Standards Stream

Maturity level 2 requires completed verification checklists and maturity level 3 requires generation of compliance reports. These checklists and compliance reports can be submitted to RABET-V with the product revision and used by the evaluators.

#### Policy and Compliance/Compliance Management Stream

Maturity level 2 requires mapping of external compliance obligations to well-defined set of application requirements. This mapping should be done for the Non-Voting Election Technology Security Best Practices. If done, maturity level 3 requires regularly reporting on compliance metrics. These reports would greatly assist and help expedite the RABET-V process.

#### Education and Guidance

Level 2 and 3 of this stream requires that the organization has established standard and guidelines relating to the Application Security, such documentation could be useful for RABET-V evaluator to determine a base level of organizational maturity.

#### Threat Assessment/Threat Modeling Stream

All maturity levels require some level of threat modeling. Threat modeling artifacts can prove useful for the initial Architecture Review and Verification activities. Changes to the threat model will also prove useful for subsequent RABET-V iterations.

#### Security Requirements/Supplier Security Stream

Maturity level 1 requires a vendor questionnaire be used to assess the strengths and weakness of suppliers. These questionnaires will be useful for the initial Architecture Review. Updated or new questionnaires will be useful for future RABET-V iterations.

Maturity level 2 requires measurement of key performance indicators for key vendor security processes. These KPI measurements would be a useful artifact for the RABET-V process.

Maturity level 3 requires verification that solution meets quality and security objectives before every major release. Evidence that this was done, such as a checklist, would be a useful artifact for the RABET-V process.

All levels of security requirements requires the creation and maintenance of a security requirements for applications. The inclusion of these security requirements and their alignment to the threats and risks identified in the threat modeling and risk assessment section would be useful for the RABET-V assessors to validate the security requirements.

#### Secure Architecture/Architecture Design Stream

All maturity levels produce artifacts which will be useful during the Architecture Review. The greater the maturity, the more useful the artifact will be to the architecture review. These will assist in expediting the architecture review and ensuring its output is an accurate as possible.

#### Secure Architecture/Technology Management Stream

Maturity level 1 requires a list of the most important technologies used in the application and their risks. This information will be useful for the initial and subsequent Architecture Reviews. Changes to this list, which must be updated at least annually for maturity level 2, might be useful to help determine when new architecture reviews are necessary.

#### Secure Build/Build Process Stream

Maturity level 3 requires that the application build fails if it doesn’t meet a predefined security baseline. Having a copy of this security baseline and the results of the build process verifying compliance with the security baseline will prove very useful for each RABET-V iteration. Depending on the content of the security baseline, it can greatly streamline the verification process.

#### Secure Build/Software Dependencies

Maturity level 1 requires the documentation of software dependencies at least once in the last 3 months. Providing updated documentation of dependencies can assist with the initial and subsequent RABET-V iterations.

Maturity level 3 requires the scanning of dependencies using a static analysis tool. This will produce results which will be useful in the RABET-V process.

#### Secure Deployment/Deployment

All maturity requires the creation and maintenance of a deployment process, this artifact and it’s inclusion of the secure means of handing secrets would be useful for RABET-V assessors. Depending on how the deployment is managed, scripts and associated outputs of the deployment process could be seen as useful artifacts.

#### Defect Management/Defect Tracking

Maturity level 2 requires SLA and tracking of compliance with SLAs. This information can assist the RABET-V process ensure that defects are being properly managed.

#### Defect Management/Metrics and Feedback

All maturity levels require recorded metrics and analysis. The higher maturity levels ensure these metrics are reliable and acted upon. Providing this data to RABET-V can be helpful depending on which metrics are being tracked and how reliable those metrics are determined to be.

#### Architecture Assessment/Architecture Validation

The overall software architecture required in maturity level 1 and other information required in maturity levels 2 and 3 will prove useful for the Architecture Review.

#### Architecture Assessment/Architecture Mitigation

None

#### Requirements Testing/Control Verification

Maturity level 2 requires the capturing of results. The extent of the test suite and results of these tests can be helpful artifacts in the RABET-V process. The more extensive and reliable these tests are, the more than can be used.

#### Requirements Testing/Misuse/Abuse Testing

Maturity level 2 requires the creation of abuse profiles and personas, such documentation which could be included as part of the threat modeling documentation would be useful for the RABET-V assessors.

#### Security Testing/Scalable Baseline

Maturity level 3 requires the tracking and reviewing of security test results. These results can be used by RABET-V to augment or substitute additional testing.

#### Security Testing/Deep Understanding

Maturity level 2 and 3 require a review of the results from testing such as penetration tests in maturity level 2. These results can be used by RABET-V to augment or substitute additional testing.

#### Incident Management

Artifacts resulting from incident detection and response will be useful for RABET-V. This including items like the root cause analysis required by maturity level 3 of the Incident Response stream and others.

#### Environment Management/Configuration Hardening

Maturity level 3 requires regular conformity checks with the hardening guidance. The results of these checks can be provided to RABET-V to offset the same effort by RABET-V.

#### Environment Management/Patching and Updating

Maturity level 3 requires a list of components and versions. This list should be submitted to the RABET-V iteration as evidence of recent patches.

#### Operational Management/Data Protection

Maturity Level 3 requires automated monitoring of attempted or actual violations of the data protection policy. Results from this monitoring can be submitted to RABET-V and assist with an assessment of security controls and response.

#### Operational Management/System Decommissioning/Legacy Management

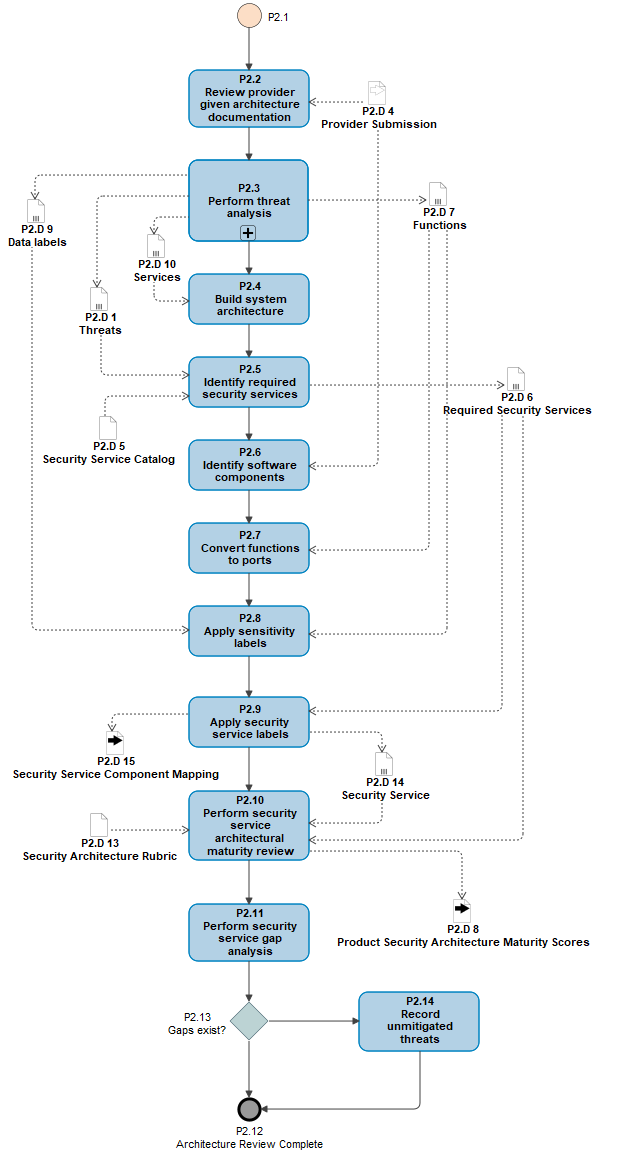
None

### Technical Guidance

* [OWASP SAMM](https://owaspsamm.org/)
* [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)
* [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 10 [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 achieves the security service (i.e. its capability level), just how mature the architecture is that supports the current capability level. 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 effect 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 maturity scores, the less testing required to validate the security capability scores.

This activity will also complete the system, security and software architectural viewpoints.

### Inputs

* Provider Submission

The Technology Provider will supply architecture diagrams, architecture descriptions, software source code, and access to a functioning version of the solution. The architecture review will use the source code and functioning solution to validate or fill-in missing pieces from the architecture diagrams and descriptions. 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.

### Outputs

* Product Security Architecture Maturity Scores

Based on the maturity scoring rubric, the architecture will be assigned a score for each security service which corresponds to how well it supports that security service.

* Security Service Component Mapping

For each security service, the Architecture Review will identify the product components at the system and software levels which *PROVIDE* and *CONFIGURE* the security service and those components which *USE* the component that provides the security service. The components which *PROVIDE* and/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). This mapping of components is referred to as the [Security Services Architecture](file:///C:\Users\wilso\RABET-V_Glossary).

### Workflow

#### P2.2 Review provider given architecture documentation

The registered technology provider will provide documentation of the product’s architecture. Use of other submitted artifacts, such as source code or a working system will be used as necessary to validate the documentation and complete the architecture review.

#### P2.3 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. 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).

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 (low, medium, high) which will influence the severity level determination of any identified threat.

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

#### P2.4 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.

#### P2.5 Identify required security services

The security services required for a given product will depend on the results of the threat modeling exercise. Security services are selected from the 10 security services identified for RABET-V.

#### P2.6 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.

Component identification will perform:

* Identification of components and boundaries
  + Boundaries between 3rd party components and election technology components
  + Boundaries between components of differing sensitivity
  + Trust boundaries

Outputs will include:

* Components - what function does the component serve, how critical is the function, how critical is the data it uses
* Boundaries - where is the logical divide between this component and another, how well-defined is the boundary, is it a trust boundary
* Dependencies - which components depend on each other, which dependencies are third parties

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

#### P2.7 Convert threat analysis to component mapping

The threat analysis identifies the software functionalities which need security services and the previous step identified the software components. This step will map the security services to software components by first identifies which components offer which functionalities.

This is done by converting the software functions from the threat model to ports on the software component diagram. (Ports are how functionality is expressed on UML component diagrams.) Here we also map the ports to software components identified in the previous step. With this combined point of view, it is then clear which software components should be providing/uses a security service.

#### P2.8 Convert threat analysis to sensitivity labels

Sensitivity labels are applied to ports that provide/exchange data-critical or data-sensitive information during the threat analysis step. Using the component to port 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.

#### P2.9 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.

#### P2.10 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)

#### P2.11 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.

#### P2.14 Record unmitigated threats

Record the unmitigated threats in the product.

#### Data used in Process

##### [] Threats

A threat is role 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., theft) as opposed to technique (e.g., spoofing). In some books and articles, the different but highly related terms “attack” and “threat” are sometimes confounded by being used as synonyms [Firesmith 03, Tulloch 03].

##### [] Security Service Catalog

A set of security services identified by RABET-V to mitigate threats.

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

##### [] Required Security Services

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

##### [] Functions

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

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

##### [] 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 Architecture Rubric

##### [] Security Service

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

## Security Claims Validation

This security claims 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/or 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 Submission Package must include claims for each security requirement. Subsequent Revision Submissions can add, remove, or modify a previous security claim, but are not required to.

For each requirement, the provider must include:

* Whether the requirement is 1. Met, 2. Partially Met, 3. Not Met, or 4. Not Applicable
* Which component or sub-systems implements the requirement, and whether it is all or a sub-set of components
* Rationale for Not Applicable (only if Not Applicable is claimed)
* Implementation details, including as much detail as possible
* 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 fully 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 Met or Not Applicable.

### Inputs

* Product goals (including in provider’s submission package)
* Product expected usage (including in provider’s submission package)
* Product security claims (including 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

#### 1. 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.

#### 2. 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.

#### 3. Review Remaining Requirements

Once the list of applicable requirements have been determined in the prior step, the reviewer will go through the remaining 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 of 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.

#### 4. Determine Claim Sufficiency

In this final step, the reviewer will analysis 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 county interact with each voter. This data is managed in the state or county voter registration systems. Depending on the state, the data flow between state and county 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 elections 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 - do not 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

#### 1. 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).

#### 2. 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 develops by technology provider
2. Software Application Supporting Component - framework, library or module provided by a 3rd party used within the software application
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) (only applies to )
* 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.

* Immature: < 1.0
* Good Maturity: 1.0-2.0
* Excellent Maturity: 2.0 -3.0

#### 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.

* Immature: < 1.0
* Good Maturity: 1.0-2.0
* Excellent Maturity: 2.0 -3.0

##### 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.

* Immature: < 1.0
* Good Maturity: 1.0-2.0
* Excellent Maturity: 2.0 -3.0

##### 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.

* Immature: < 1.0
* Good Maturity: 1.0-2.0
* Excellent Maturity: 2.0 -3.0

##### 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.

* Immature: < 1.0
* Good Maturity: 1.0-2.0
* Excellent Maturity: 2.0 -3.0

##### Software Development Artifacts

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

Is there a reliable artifact available? Possible 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 calculated by 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

#### 1. 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.

#### 2. 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.

#### 3. 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. During the initial RABET-V iteration, sanity testing will be performed against a 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 reviewers 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 Verification Authority 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. 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 the 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

#### 1. 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

# SSCM Model

The SSCM Index provides maturity scores across 10 security services. The pages below detail what each maturity level entails.

## Authentication Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of authentication best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 authentication best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authentication

### Maturity Level 2

#### Benefit

Advanced level of authentication best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 authentication best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authentication

### Maturity Level 3

#### Benefit

Optimal level of authentication best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 authentication best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authentication

## Authorization Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of authorization best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 authorization best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authorization

### Maturity Level 2

#### Benefit

Advanced level of authorization best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 authorization best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authorization

### Maturity Level 3

#### Benefit

Optimal level of authorization best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 authorization best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require authorization

## Boundary Protection Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of boundary protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 boundary protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some boundaries
* Yes, on most boundaries and/or they are provided manually
* Yes, on all boundaries that require boundary protection

### Maturity Level 2

#### Benefit

Advanced level of boundary protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 boundary protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some boundaries
* Yes, on most boundaries and/or they are provided manually
* Yes, on all boundaries that require boundary protection

### Maturity Level 3

#### Benefit

Optimal level of boundary protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 boundary protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some boundaries
* Yes, on most boundaries and/or they are provided manually
* Yes, on all boundaries that require boundary protection

## Data Confidentiality Protection Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of data confidentiality protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 data confidentiality protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data confidentiality protection

### Maturity Level 2

#### Benefit

Advanced level of data confidentiality protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 data confidentiality protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data confidentiality protection

### Maturity Level 3

#### Benefit

Optimal level of data confidentiality protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 data confidentiality protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data confidentiality protection

## Data Integrity Protection Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of data integrity protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 data integrity protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data integrity protection

### Maturity Level 2

#### Benefit

Advanced level of data integrity protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 data integrity protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data integrity protection

### Maturity Level 3

#### Benefit

Optimal level of data integrity protection best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 data integrity protection best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require data integrity protection

## Injection Prevention Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of injection prevention best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 injection prevention best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require injection prevention

### Maturity Level 2

#### Benefit

Advanced level of injection prevention best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 injection prevention best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require injection prevention

### Maturity Level 3

#### Benefit

Optimal level of injection prevention best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 injection prevention best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require injection prevention

## Logging and Alerting Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of logging and alerting best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 logging and alerting best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require logging and alerting

### Maturity Level 2

#### Benefit

Advanced level of logging and alerting best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 logging and alerting best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require logging and alerting

### Maturity Level 3

#### Benefit

Optimal level of logging and alerting best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 logging and alerting best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require logging and alerting

## Secret Management Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of secret management best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 secret management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require secret management

### Maturity Level 2

#### Benefit

Advanced level of secret management best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 secret management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require secret management

### Maturity Level 3

#### Benefit

Optimal level of secret management best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 secret management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require secret management

## System Integrity Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of system integrity best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 system integrity best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require system integrity

### Maturity Level 2

#### Benefit

Advanced level of system integrity best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 system integrity best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require system integrity

### Maturity Level 3

#### Benefit

Optimal level of system integrity best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 system integrity best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require system integrity

## User Session Management Capability Maturity

### Maturity Level 1

#### Benefit

Basic level of user session management best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 1 user session management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require user session management

### Maturity Level 2

#### Benefit

Advanced level of authentication best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 2 user session management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require user session management

### Maturity Level 3

#### Benefit

Optimal level of user session management best practices are implemented across the product in order to ensure only valid users are given access.

#### Quality criteria

To be filled in with profile level 3 user session management best practices.

#### Question

* Do you implement these technology controls or activities?
* Does the product provide these features or controls?

#### Answers

* No
* Yes, on some components
* Yes, on most components and/or they are provided manually
* Yes, on all components that require user session management

# SSAM Model

The SSAM Index provides maturity scores across two streams.

* [Security Service Construction](file:///C:\Users\wilso\Downloads\SSAMModel\Security_Service_Construction_Maturity)
* [Security Service Usage](file:///C:\Users\wilso\Downloads\SSAMModel\Security_Service_Usage_Maturity)

## Security Service Construction Maturity

### Maturity Level 1

#### Benefit

The security service is being provided through well-vetted components.

#### Quality Criteria

* The trustworthiness and reliability of the security service components have been vetted

#### Discussion

In the vast majority of cases, the security service components should be written by a reputable third party. Use of security service components developed or maintained in-house must be thoroughly justified.

#### Question

* Who wrote or provides the security service components?
* Was it written in house?
* Was it provided by a reputable third party?

#### Answers

* No, security service not vetted (no credit)
* Yes, written in house (partial credit)
* Yes, written by third party (full credit)

### Maturity Level 2

#### Benefit

The security service components are well documented.

#### Quality criteria

* Proper usage of the security service component can be determined from documentation
* Illustrative examples of the security service are provided

#### Questions

* Is the security service well documented?

#### Answers

* No documentation (no credit)
* Limited, inadequate documentation (partial credit)
* Comprehensively documented (full credit)

### Maturity Level 3

#### Benefit

There is a consistent and stable interface that integrates well with the chosen technology framework.

#### Quality Criteria

* The security service API is clear and consistent
* The API provides required features in a straightforward manner (i.e. no unnecessary complexity)
* The security service provides interfaces that work with chosen technology stack

#### Discussion

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 no provide extension points for a given security service will not be penalized.

#### Questions

* Does the security service provide a consistent interface?
* Does the interface integrate with the chosen technology stack?

#### Answers

* No (no credit)
* Yes, but not integrated with available framework extension point (partial credit)
* Yes, is integrated with underlying framework (full credit)

## Security Service Usage Maturity

### Maturity Level 1

#### Benefit

A dedicated component providing a security service is used within the subsystem.

#### Quality criteria

There is a minimal number of components which provide the security service. This is ideally one component per technical stack or subsystem. Use of multiple components providing the same mitigations must be justified.

#### Discussion

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.

#### Questions

* Is there an isolated, dedicated component providing the security service for the subsystem?

#### Answers

* Multiple components are involved in providing the security system in a subsystem (no credit)
* One component is used but it is not dedicated to security mitigations (partial credit)
* One dedicated component is used to provide the security system within the subsystem (full credit)

### Maturity Level 2 (Centralized configuration)

#### Benefit

The security service is centrally configured.

#### Quality Criteria

* The security service is configured and used consistently throughout the subsystem.
* There is one component which configures the security service provider.

#### Discussion

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.

#### Questions

* What component(s) configure the security service?
* Is the centralized configuration overridden by other components?

#### Answer

* Usually or always configured at the point of use (no credit)
* Mostly configured centrally, with some service details configured at point of use (partial credit)
* Configured centrally, or no configuration required (full credit)

### Maturity Level 3

#### Benefit

The security service is used consistently and appropriately throughout the system and it is easy to verify its use.

#### Quality Criteria

* The component(s) providing the security service is used by other components throughout the system, where appropriate
* Reusable components are constructed to make is easy to use the security service
* System defaults encourage or enforce the use of the security service

#### Discussion

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.

#### Questions

* How widespread and appropriate is the use of the security service?
* Are there defaults or reusable components which make it easy to use and verify the use of the security service?

#### Answers

* There is little to no consistent use of the security service or it’s use is missing from key components (no credit)
* There is consistent use of the security service throughout the application but it’s use is not the default or easily verifiable (partial credit)
* There is consistent use of the security service which is made easy and verifiable by defaults and/or reusable components (full credit)

# SDM Model

The SDM Model is based on OWASP SAMM. The links below will take you to the relevant pages in OWASP SAMM’s website. Accessibility and Usability were defined specifically for RABET-V and those maturity model details are provided below.

## [Strategy & Metrics](https://owaspsamm.org/model/governance/strategy-and-metrics/)

* [Create and Promote](https://owaspsamm.org/model/governance/strategy-and-metrics/stream-a/)
* [Measure and Improve](https://owaspsamm.org/model/governance/strategy-and-metrics/stream-b/)

## [Policy & Compliance](https://owaspsamm.org/model/governance/policy-and-compliance/)

* [Policy & Standards](https://owaspsamm.org/model/governance/policy-and-compliance/stream-a/)
* [Compliance Management](https://owaspsamm.org/model/governance/policy-and-compliance/stream-b/)

## [Education & Guidance](https://owaspsamm.org/model/governance/education-and-guidance)

* [Training and Awareness](https://owaspsamm.org/model/governance/education-and-guidance/stream-a/)
* [Organization and Culture](https://owaspsamm.org/model/governance/education-and-guidance/stream-b/)

## [Threat Assessment](https://owaspsamm.org/model/design/threat-assessment)

* [Application Risk Profile](https://owaspsamm.org/model/design/threat-assessment/stream-a/)
* [Threat Modeling](https://owaspsamm.org/model/design/threat-assessment/stream-b/)

## [Security Requirements](https://owaspsamm.org/model/design/security-requirements)

* [Software Requirements](https://owaspsamm.org/model/design/security-requirements/stream-a/)
* [Supplier Security](https://owaspsamm.org/model/design/security-requirements/stream-b/)

## [Secure Architecture](https://owaspsamm.org/model/design/security-architecture)

* [Architecture Design](https://owaspsamm.org/model/design/security-architecture/stream-a/)
* [Technology Management](https://owaspsamm.org/model/design/security-architecture/stream-b/)

## [Secure Build](https://owaspsamm.org/model/implementation/secure-build)

* [Build Process](https://owaspsamm.org/model/implementation/secure-build/stream-a/)
* [Software Dependencies](https://owaspsamm.org/model/implementation/secure-build/stream-b/)

## [Secure Deployment](https://owaspsamm.org/model/implementation/secure-deployment)

* [Deployment Process](https://owaspsamm.org/model/implementation/secure-deployment/stream-a/)
* [Secret Management](https://owaspsamm.org/model/implementation/secure-deployment/stream-b/)

## [Defect Management](https://owaspsamm.org/model/implementation/defect-management)

* [Defect Tracking](https://owaspsamm.org/model/implementation/defect-management/stream-a/)
* [Metrics and Feedback](https://owaspsamm.org/model/implementation/defect-management/stream-b/)

## [Architecture Assessment](https://owaspsamm.org/model/verification/architecture-assessment)

* [Architecture Validation](https://owaspsamm.org/model/verification/architecture-assessment/stream-a/)
* [Architecture Mitigation](https://owaspsamm.org/model/verification/architecture-assessment/stream-b/)

## [Requirements Testing](https://owaspsamm.org/model/verification/requirements-driven-testing)

* [Control Verification](https://owaspsamm.org/model/verification/requirements-driven-testing/stream-a/)
* [Misuse/Abuse Testing](https://owaspsamm.org/model/verification/requirements-driven-testing/stream-b/)

## [Security Testing](https://owaspsamm.org/model/verification/security-testing)

* [Scalable Baseline](https://owaspsamm.org/model/verification/security-testing/stream-a/)
* [Deep Understanding](https://owaspsamm.org/model/verification/security-testing/stream-b/)

## [Incident Management](https://owaspsamm.org/model/operations/incident-management)

* [Incident Detection](https://owaspsamm.org/model/operations/incident-management/stream-a/)
* [Incident Response](https://owaspsamm.org/model/operations/incident-management/stream-b/)

## [Environment Management](https://owaspsamm.org/model/operations/environment-management)

* [Configuration Hardening](https://owaspsamm.org/model/operations/environment-management/stream-a/)
* [Patching and Updating](https://owaspsamm.org/model/operations/environment-management/stream-b/)

## [Operational Management](https://owaspsamm.org/model/operations/operational-management)

* [Data Protection](https://owaspsamm.org/model/operations/operational-management/stream-a/)
* [System Decomissioning / Legacy Management](https://owaspsamm.org/model/operations/operational-management/stream-b/)

## Accessibility

### Maturity level 1

#### Benefit

Automated conformance to accessibility guidelines

#### Activity

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.

Automated accessibility testing ensure conformance to the accessibility standards. The actual implementation often falls short of true accessibility; however, it is the first step.

#### Question

Do you perform automated accessibility validation during development?

#### Quality criteria

* You prioritize meeting accessibility standards as an organization
* You use automated accessibility testing tools as a part of your development processes

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

### Maturity level 2

#### Benefit

Better real-world accessibility

#### Activity

Test the solution with accessibility tools, such as OS-specific features or commercial hardware and software.

#### Question

Do you perform accessibility tests with commercial accessibility software and OS-specific features?

#### Quality criteria

* You prioritize creating accessible solutions as an organization
* You create personas and scenarios to aid in functionally testing the solution using accessibility software

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

### Maturity level 3

#### Benefit

Functionally accessible solution for a wide-range of abilities and stronger relationships with the accessibility community

#### Activity

Accessibility experts perform formal research with users that have various accessibility needs. This involves the users interacting with a prototype or version of the software solution.

The accessibility experts develop formal reports on their findings and share them with the product teams.

#### Question

Do you employ or contract with accessibility experts to perform accessibility testing and analysis?

#### Quality criteria

* You employ or contract with accessibility experts
* You perform formal accessibility studies
* You work with members of various communities to account for a wide range of abilities
* You integrate the findings of these studies into the product development

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

## Usability

### Maturity level 1

#### Benefit

Software solution more closely aligns with an organization’s processes

#### Activity

Whether the process is momentary, ad hoc, or iterative, requirements gathering is a substantial part of software development. Requirements gathering alone is insufficient to create an application that accurately fits the needs of the people that use it. Usability testing and analysis helps bridge the gap between an solution that meets a set of requirements and a solution that meets the needs of the organization, people, and processes. This distinction is really between an solution that people want to use—i.e. one that has been usability tested—versus one they don’t—i.e. one that simply meets a set of requirements. Poorly-designed applications can result in the loss of process information, the collection of incorrect or malformed data, and/or cause users to look for alternative means of performing their work outside the technology. Any of these issues may introduce risk into the organization or inhibit the diagnosis of a security event.

Even if all requirements are met during a development phase of a software solution, more valuable feedback is gained after it’s used in a production environment. Survey the customers for feedback on how the product is working.

#### Question

Do you have a formal feedback loop with your customers?

#### Quality criteria

* You issue a survey about the application with the customer
* You use the results of the survey to refine the solution
* You perform this activity after every deployment

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

### Maturity level 2

#### Benefit

The application conforms to the needs of the people using the solution

#### Activity

Work with the customer to interview users of the software solution to find out their processes and pain points.

If the solution is internet connected, set up a direct feedback form from within the application. If possible, collect logs and analytics from the system.

Produce formal reports of these findings and review the analysis with the product teams.

#### Question

Do you work to collect direct feedback from users and perform analysis on the findings?

#### Quality criteria

* You work with the customer to talk directly to the users of the solution to find out what they do, their processes, how they use the system, what they appreciate about the tool, and what they feel could be added or improved
* You collect usage data directly from the system through logs or analytics
* You perform analysis of the information gathered and share it with the product team

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

### Maturity level 3

#### Benefit

The application conforms to the users’ behaviors and business processes

#### Activity

Usability experts perform formal research on the business processes, the users’ behaviors, and conduct usability studies with users interacting with a prototype or version of the software solution.

The usability experts develop formal reports on their findings and share them with the product teams.

#### Question

Do you employ or contract with usability experts to perform usability testing and analysis?

#### Quality criteria

* You employ or contract with usability experts
* You perform formal usability studies
* You integrate the findings of these studies into the product development

#### Answers

* No
* Yes, for some applications
* Yes, for at least half the applications
* Yes, for most or all of the applications

# 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.

### 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

### 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.

### process

### product

an election technology submitt to RABET-V.

### product revision

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

### 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.

### 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.

### verification authority

The organization responsible for overseeing and executing the RABET-V [process](#process). CIS is the Verification Authority for the pilot program.