# 



# Application Security Verification Standard 3.1

[date]

# Frontispiece

## About the Standard

Welcome to the Mobile Application Security Verification Standard (MASVS) 0.9.2. The MASVS is a community effort to establish a framework of security requirements needed to design, develop and test secure mobile apps on iOS and Android.

The MASVS is a culmination of community effort and industry feedback. We expect this standard to evolve over time and welcome feedback from the community. The best way to get in contact with us is via the OWASP Mobile Project Slack channel:

<https://owasp.slack.com/messages/project-mobile_omtg/details/>

Accounts can be created at the following URL:

<http://owasp.herokuapp.com/>.

## Copyright and License

  
Copyright © 2017 The OWASP Foundation. This document is released under the Creative Commons Attribution ShareAlike 3.0 license. For any reuse or distribution, you must make clear to others the license terms of this work.

|  |  |  |
| --- | --- | --- |
| Project Leads | Lead Authors | Contributors and Reviewers |
| Bernhard Mueller, Sven Schleier | Bernhard Mueller | Stephen Corbiaux, Sven Schleier, Jeroen Willemsen, Anant Shrivastava, Abdessamad Temmar, Alexander Antukh, Roberto Martelloni, Stefaan Seys, Prabhant Singh, Francesco Stillavato, Abhinav Sejpal |

This document started as a fork of the OWASP Application Security Verification Standard written by Jim Manico.

# Preface

Welcome to the Application Security Verification Standard (ASVS) version 3.0. The ASVS is a community-effort to establish a framework of security requirements and controls that focus on normalising the functional and non-functional security controls required when designing, developing and testing modern web applications.  
ASVS v3.0 is a culmination of community effort and industry feedback. In this release, we felt it was important to qualify the experiences of real world use cases relating to ASVS adoption. This will help newcomers to the standard plan their adoption of the ASVS, whilst assisting existing companies in learning from the experience of others.

We expect that there will most likely never be 100% agreement on this standard. Risk analysis is always subjective to some extent, which creates a challenge when attempting to generalize in a one size fits all standard. However, we hope that the latest updates made in this version are a step in the right direction, and respectfully enhance the concepts introduced in this important industry standard.

## What’s new in 3.0?

In version 3.0, we have added several new sections, including Configuration, Web Services, Modern (Client) based applications, and IoT Devices to make the Standard more applicable to modern applications, which are commonly responsive applications, with an extensive HTML5 front end or mobile client that calls a common set of RESTful web services using SAML authentication.

We have also de-duplicated the standard, for example, to ensure that a mobile developer does not need to re-test the same items multiple times.

We have provided a mapping to the CWE common weakness enumeration (CWE) dictionary. The CWE mapping can be used to identify information such as likelihood of exploitation, consequence of a successful exploitation and broadly speaking to gain insight on what could go wrong if a security control is not used or implemented effectively and how to mitigate the weakness.

Lastly, we reached out to the community and held peer review sessions at AppSec EU 2015 and a final working session at AppSec USA 2015 to include a massive amount of community feedback. During peer review, if edits to the meaning of a control changed substantially, we created a new control and deprecated the old one. We have deliberately chosen to not reuse any deprecated control requirements, as this could be a source of confusion. We have provided a comprehensive mapping of what has changed in Appendix A.

Taken together, v3.0 is the single largest change to the Standard in its history. We hope that you find the update to the standard useful, and use it in ways we can only imagine.

# Using the ASVS

ASVS has two main goals:

* to help organizations develop and maintain secure applications
* to allow security service, security tools vendors, and consumers to align their requirements and offerings

Application Security Verification Levels  
The Application Security Verification Standard defines three security verification levels, with each level increasing in depth.

* ASVS Level 1 is meant for all software.
* ASVS Level 2 is for applications that contain sensitive data, which requires protection.
* ASVS Level 3 is for the most critical applications - applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

Each ASVS level contains a list of security requirements. Each of these requirements can also be mapped to security-specific features and capabilities that must be built into software by developers.

Figure 1 - OWASP Application Security Verification Standard 3.0 Levels

## How to use this standard

One of the best ways to use the Application Security Verification Standard is to use it as blueprint create a Secure Coding Checklist specific to your application, platform or organization. Tailoring the ASVS to your use cases will increase the focus on the security requirements that are most important to your projects and environments.

### Level 1: Opportunistic

An application achieves ASVS Level 1 (or Opportunistic) if it adequately defends against application security vulnerabilities that are easy to discover, and included in the OWASP Top 10 and other similar checklists.

Level 1 is typically appropriate for applications where low confidence in the correct use of security controls is required, or to provide a quick analysis of a fleet of enterprise applications, or assisting in developing a prioritized list of security requirements as part of a multi-phase effort. Level 1 controls can be ensured either automatically by tools or simply manually without access to source code. We consider Level 1 the minimum required for all applications.

Threats to the application will most likely be from attackers who are using simple and low effort techniques to identify easy-to-find and easy-to-exploit vulnerabilities. This is in contrast to a determined attacker who will spend focused energy to specifically target the application. If data processed by your application has high value, you would rarely want to stop at a Level 1 review.

### Level 2: Standard

An application achieves ASVS Level 2 (or Standard) if it adequately defends against most of the risks associated with software today.  
Level 2 ensures that security controls are in place, effective, and used within the application. Level 2 is typically appropriate for applications that handle significant business-to-business transactions, including those that process healthcare information, implement business-critical or sensitive functions, or process other sensitive assets.  
Threats to Level 2 applications will typically be skilled and motivated attackers focusing on specific targets using tools and techniques that are highly practiced and effective at discovering and exploiting weaknesses within applications.

### Level 3: Advanced

ASVS Level 3 is the highest level of verification within the ASVS. This level is typically reserved for applications that require significant levels of security verification, such as those that may be found within areas of military, health and safety, critical infrastructure, etc.

Organisations may require ASVS Level 3 for applications that perform critical functions, where failure could significantly impact the organization's operations, and even its survivability. Example guidance on the application of ASVS Level 3 is provided below. An application achieves ASVS Level 3 (or Advanced) if it adequately defends against advanced application security vulnerabilities and also demonstrates principles of good security design.  
An application at ASVS Level 3 requires more in depth analysis, architecture, coding, and testing than all the other levels. A secure application is modularized in a meaningful way (to facilitate e.g. resiliency, scalability, and most of all, layers of security), and each module (separated by network connection and/or physical instance) takes care of its own security responsibilities (defence in depth), that need to be properly documented. Responsibilities include controls for ensuring confidentiality (e.g. encryption), integrity (e.g. transactions, input validation), availability (e.g. handling load gracefully), authentication (including between systems), non-repudiation, authorization, and auditing (logging).

## Applying ASVS in Practice

Different threats have different motivations. Some industries have unique information and technology assets and domain specific regulatory compliance requirements.

Below we provide industry-specific guidance regarding recommended ASVS levels. Although some unique criteria and some differences in threats exist for each industry, a common theme throughout all industry segments is that opportunistic attackers will look for any easily exploitable vulnerable applications, which is why ASVS Level 1 is recommended for all applications regardless of industry. This is a suggested starting point to manage the easiest to find risks. Organizations are strongly encouraged to look more deeply at their unique risk characteristics based on the nature of their business. At the other end of the spectrum is ASVS Level 3, which is reserved for those cases that might endanger human safety or when a full application breach could severely impact the organization.

| Industry | Threat Profile | L1 Recommendation | L2 Recommendation | L3 Recommendation |  
| -- | -- | -- | -- | -- |  
| Finance and Insurance | Although this segment will experience attempts from opportunistic attackers, it is often viewed as a high value target by motivated attackers and attacks are often financially motivated. Commonly, attackers are looking for sensitive data or account credentials that can be used to commit fraud or to benefit directly by leveraging money movement functionality built into applications. Techniques often include stolen credentials, application-level attacks, and social engineering. Some major compliance considerations include Payment Card Industry Data Security Standard (PCI DSS),Gramm Leech Bliley Act and  
Sarbanes-Oxley Act (SOX). | Although this segment will experience attempts from opportunistic attackers, it is often viewed as a high value target by motivated attackers and attacks are often financially motivated. Commonly, attackers are looking for sensitive data or account credentials that can be used to commit fraud or to benefit directly by leveraging money movement functionality built into applications. Techniques often include stolen credentials, application-level attacks, and social engineering. Some major compliance considerations include Payment Card Industry Data Security Standard (PCI DSS),Gramm Leech Bliley Act and  
Sarbanes-Oxley Act (SOX). All network accessible applications. | Applications that contain sensitive information like credit card numbers, personal information, that can move limited amounts of money in limited ways. Examples include:

* transfer money between accounts at the same institution or
* a slower form of money movement (e.g. ACH) with transaction limits or
* wire transfers with hard transfer limits within a period of time. | Applications that contain sensitive information like credit card numbers, personal information, that can move limited amounts of money in limited ways. Examples include:
* transfer money between accounts at the same institution or
* a slower form of money movement (e.g. ACH) with transaction limits or
* wire transfers with hard transfer limits within a period of time. Applications that contain large amounts of sensitive information or that allow either rapid transfer of large sums of money (e.g. wire transfers) and/or transfer of large sums of money in the form of individual transactions or as a batch of smaller transfers. |

# Assessment and Certification

## OWASP's Stance on MASVS Certifications and Trust Marks

OWASP, as a vendor-neutral not-for-profit organization, does not certify any vendors, verifiers or software.

All such assurance assertions, trust marks, or certifications are not officially vetted, registered, or certified by OWASP, so an organization relying upon such a view needs to be cautious of the trust placed in any third party or trust mark claiming ASVS certification.

This should not inhibit organizations from offering such assurance services, as long as they do not claim official OWASP certification.

## Guidance for Certifying Mobile Apps

The recommended way of verifying compliance of a mobile app with the MASVS is by performing an "open book" review, meaning that the testers are granted access to key resources such as architects and developers of the app, project documentation, source code, and authenticated access to endpoints, including access to at least one user account for each role.

It is important to note that the MASVS only covers security of the (client-side) mobile app and the network communication between the app and its remote endpoint(s), as well as a few basic and generic requirements related to user authentication and session management. It does not contain specific requirements for the remote services (e.g. web services) associated with the app, safe for a limited set of generic requirements pertaining to authentication and session management. However, MASVS V1 specifies that remote services must be covered by the overall threat model, and be verified against appropriate standards, such as the OWASP ASVS.

A certifying organization must include in any report the scope of the verification (particularly if a key component is out of scope), a summary of verification findings, including passed and failed tests, with clear indications of how to resolve the failed tests. Keeping detailed work papers, screenshots or movies, scripts to reliably and repeatedly exploit an issue, and electronic records of testing, such as intercepting proxy logs and associated notes such as a cleanup list, is considered standard industry practice. It is not sufficient to simply run a tool and report on the failures; this does not provide sufficient evidence that all issues at a certifying level have been tested and tested thoroughly. In case of dispute, there should be sufficient supportive evidence to demonstrate that every verified requirement has indeed been tested.

### Using the OWASP Mobile Security Testing Guide (MSTG)

The OWASP MSTG is a manual for testing the security of mobile apps. It describes the technical processes for verifying the requirements listed in the MASVS. The MSTG includes a list of test cases, each of which map to a requirement in the MASVS. While the MASVS requirements are high-level and generic, the MSTG provides in-depth recommendations and testing procedures on a per-mobile-OS basis.

### The Role of Automated Security Testing Tools

The use of source code scanners and black-box testing tools is encouraged in order to increase efficiency whenever possible. It is however not possible to complete MASVS verification using automated tools alone: Every mobile app is different, and understanding the overall architecture, business logic, and technical pitfalls of the specific technologies and frameworks being used, is a mandatory requirement to verify security of the app.

## Other Uses

### As Detailed Security Architecture Guidance

One of the more common uses for the Mobile Application Security Verification Standard is as a resource for security architects. The two major security architecture frameworks, SABSA or TOGAF, are missing a great deal of information that is necessary to complete mobile application security architecture reviews. MASVS can be used to fill in those gaps by allowing security architects to choose better controls for issues common to mobile apps.

### As a Replacement for Off-the-shelf Secure Coding Checklists

Many organizations can benefit from adopting the MASVS, by choosing one of the two levels, or by forking MASVS and changing what is required for each application's risk level in a domain-specific way. We encourage this type of forking as long as traceability is maintained, so that if an app has passed requirement 4.1, this means the same thing for forked copies as the standard evolves.

### As a Basis for Security Testing Methodologies

A good mobile app security testing methodology should cover all requirements listed in the MASVS. The OWASP Mobile Security Testing Guide (MSTG) describes black-box and white-box test cases for each verification requirement.

### As a Guide for Automated Unit and Integration Tests

The MASVS is designed to be highly testable, with the sole exception of architectural requirements. Automated unit, integration and acceptance testing based on the MASVS requirements can be integrated in the continuous development lifecycle. This not only increases developer security awareness, but also improves the overall quality of the resulting apps, and reduces the amount of findings during security testing in the pre-release phase.

### For Secure Development Training

MASVS can also be used to define characteristics of secure mobile apps. Many "secure coding" courses are simply ethical hacking courses with a light smear of coding tips. This does not help developers. Instead, secure development courses can use the MASVS, with a strong focus on the proactive controls documented in the MASVS, rather than e.g. the Top 10 code security issues.

# V1: Architecture, Design and Threat Modeling Requirements

## Control Objective

In a perfect world, security would be considered throughout all phases of development. In reality however, security is often only a consideration at a late stage in the SDLC. Besides the technical controls, the ASVS requires processes to be in place that ensure that the security has been explicitly addressed when planning the architecture of the application or API, and that the functional and security roles of all components are known. Since single page applications and act as clients to remote API or services, it must be ensured that appropriate security standards are also applied to those services - testing the app in isolation is not sufficient.

The category “V1” lists requirements pertaining to architecture and design of the app. As such, this is the only category that does not map to technical test cases in the OWASP Testing Guide. To cover topics such as threat modelling, secure SDLC, key management, users of the ASVS should consult the respective OWASP projects and/or other standards such as the ones linked below.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **1.1** | All app components are identified and known to be needed. | ✓ | ✓ | ✓ | 1.0 |
| **1.2** | Security controls are never enforced only on the client side, but on the respective remote endpoints. |  | ✓ | ✓ | 1.0 |
| **1.3** | A high-level architecture for the application and all connected remote services has been defined and security has been addressed in that architecture. |  | ✓ | ✓ | 1.0 |
| **1.4** | Data considered sensitive in the context of the application is clearly identified. |  |  | ✓ | 1.0 |
| **1.5** | All app components are defined in terms of the business functions and/or security functions they provide. |  |  | ✓ | 1.0 |
| **1.6** | A threat model for the application and the associated remote services has been produced that identifies potential threats and countermeasures. |  |  | ✓ | 1.0 |
| **1.7** | All security controls have a centralized implementation. |  | ✓ | ✓ | 3.0 |
| **1.8** | Components are segregated from each other via a defined security control, such as network segmentation, firewall rules, or cloud based security groups. |  | ✓ | ✓ | 3.0 |
| **1.9** | A mechanism for enforcing updates of the application exists. |  | ✓ | ✓ | 3.0 |
| **1.10** | Security is addressed within all parts of the software development lifecycle. |  | ✓ | ✓ | 3.0 |
| **1.11** | all application components, libraries, modules, frameworks, platform, and operating systems are free from known vulnerabilities |  | ✓ | ✓ | 3.0.1 |
| **1.12** | There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57. |  | ✓ | ✓ | 3.1 |

## References

For more information, see also:

For more information, please see:

* [OWASP Threat Modeling Cheat Sheet](https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet)
* [OWASP Attack Surface Analysis Cheat Sheet](https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet)
* [OWASP Security Architecture Cheat Sheet](https://www.owasp.org/index.php/Application_Security_Architecture_Cheat_Sheet)
* [OWASP Thread modelling](https://www.owasp.org/index.php/Application_Threat_Modeling)
* [OWASP Secure SDLC Cheat Sheet](https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet)
* [Microsoft SDL](https://www.microsoft.com/en-us/sdl/)
* [NIST SP 800-57](http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf)

# V2: Authentication Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V3: Session Management Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V4: Access Control Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V5: Input Validation and Output Encoding Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V7: Cryptography Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V8: Error Handling and Logging Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V9: Data Protection Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V10: Communications Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V11: HTTP Security Configuration Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V13: Malicious Code Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V15: Business Logic Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V16: File and Resources Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V17: Mobile Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V18: API and Web Service Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V19: Configuration Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 1.0 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# V20: Internet of Things Verification Requirements

## Control Objective

TBA.

## Security Verification Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Description | L1 | L2 | L3 | Since |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |
| **TBA** | TBA | ✓ | ✓ | ✓ | 3.1 |

## References

For more information, see also:

* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA
* TBA

# Appendix A: Glossary

* **2FA** – Two-factor authentication(2FA) adds a second level of authentication to an account log-in.
* **Address Space Layout Randomization (ASLR)** – A technique to make exploiting memory corruption bugs more difficult.
* **Application Security** – Application-level security focuses on the analysis of components that comprise the application layer of the Open Systems Interconnection Reference Model (OSI Model), rather than focusing on for example the underlying operating system or connected networks.
* **Application Security Verification** – The technical assessment of an application against the OWASP MASVS.
* **Application Security Verification Report** – A report that documents the overall results and supporting analysis produced by the verifier for a particular application.
* **Authentication** – The verification of the claimed identity of an application user.
* **Automated Verification** – The use of automated tools (either dynamic analysis tools, static analysis tools, or both) that use vulnerability signatures to find problems.
* **Black box testing** – It is a method of software testing that examines the functionality of an application without peering into its internal structures or workings.
* **Component** – a self-contained unit of code, with associated disk and network interfaces that communicates with other components.
* **Cross-Site Scripting** (XSS) – A security vulnerability typically found in web applications allowing the injection of client-side scripts into content.
* **Cryptographic module** – Hardware, software, and/or firmware that implements cryptographic algorithms and/or generates cryptographic keys.
* **CWE** - CWE is a community-developed list of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.
* **DAST** –Dynamic application security testing (DAST) technologies are designed to detect conditions indicative of a security vulnerability in an application in its running state.
* **Design Verification** – The technical assessment of the security architecture of an application.
* **Dynamic Verification** – The use of automated tools that use vulnerability signatures to find problems during the execution of an application.
* **Globally Unique Identifier** (GUID) – a unique reference number used as an identifier in software.
* **Hyper Text Transfer Protocol** (HTTP) – An application protocol for distributed, collaborative, hypermedia information systems. It is the foundation of data communication for the World Wide Web.
* **Hardcoded keys** – Cryptographic keys which are stored in the device itself.
* **IPC** – Inter Process Communications,In IPC Processes communicate with each other and with the kernel to coordinate their activities.
* **Input Validation** – The canonicalization and validation of untrusted user input.
* **JAVA Bytecode** - Java bytecode is the instruction set of the Java virtual machine(JVM). Each bytecode is composed of one, or in some cases two bytes that represent the instruction (opcode), along with zero or more bytes for passing parameters.
* **Malicious Code** – Code introduced into an application during its development unbeknownst to the application owner, which circumvents the application's intended security policy. Not the same as malware such as a virus or worm!
* **Malware** – Executable code that is introduced into an application during runtime without the knowledge of the application user or administrator.
* **Open Web Application Security Project** (OWASP) – The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software. Our mission is to make application security "visible," so that people and organizations can make informed decisions about application security risks. See: <http://www.owasp.org/>
* **Personally Identifiable Information** (PII) - is information that can be used on its own or with other information to identify, contact, or locate a single person, or to identify an individual in context.
* **PIE** – Position-independent executable (PIE) is a body of machine code that, being placed somewhere in the primary memory, executes properly regardless of its absolute address.
* **PKI** – A PKI is an arrangement that binds public keys with respective identities of entities. The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA).
* **SAST** – Static application security testing (SAST) is a set of technologies designed to analyze application source code, byte code and binaries for coding and design conditions that are indicative of security vulnerabilities. SAST solutions analyze an application from the “inside out” in a nonrunning state.
* **SDLC** – Software development lifecycle.
* **Security Architecture** – An abstraction of an application's design that identifies and describes where and how security controls are used, and also identifies and describes the location and sensitivity of both user and application data.
* **Security Configuration** – The runtime configuration of an application that affects how security controls are used.
* **Security Control** – A function or component that performs a security check (e.g. an access control check) or when called results in a security effect (e.g. generating an audit record).
* **SQL Injection (SQLi)** – A code injection technique used to attack data driven applications, in which malicious SQL statements are inserted into an entry point.
* **SSO Authentication** – Single Sign On(SSO) occurs when a user logs in to one Client and is then signed in to other Clients automatically, regardless of the platform, technology, or domain the user is using. For example when you log in in google you automatically login in the youtube , docs and mail service.
* **Threat Modeling** - A technique consisting of developing increasingly refined security architectures to identify threat agents, security zones, security controls, and important technical and business assets.
* **Transport Layer Security** – Cryptographic protocols that provide communication security over the Internet
* **URI/URL/URL fragments** – A Uniform Resource Identifier is a string of characters used to identify a name or a web resource. A Uniform Resource Locator is often used as a reference to a resource.
* **User acceptance testing (UAT)**– Traditionally a test environment that behaves like the production environment where all software testing is performed before going live.
* **Verifier** – The person or team that is reviewing an application against the OWASP ASVS requirements.
* **Whitelist** – A list of permitted data or operations, for example a list of characters that are allowed to perform input validation.
* **X.509 Certificate** – An X.509 certificate is a digital certificate that uses the widely accepted international X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the user, computer or service identity contained within the certificate.

# Appendix B: References

The following OWASP projects are most likely to be useful to users/adopters of this standard:

* OWASP Mobile Security Project - <https://www.owasp.org/index.php/OWASP_Mobile_Security_Project>
* OWASP Mobile Security Testing Guide - <https://www.owasp.org/index.php/OWASP_Mobile_Security_Testing_Guide>
* OWASP Mobile Top 10 Risks - <https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks>
* OWASP Reverse Engineering and Code Modification Prevention - <https://www.owasp.org/index.php/OWASP_Reverse_Engineering_and_Code_Modification_Prevention_Project>

Similarly, the following web sites are most likely to be useful to users/adopters of this standard:

* MITRE Common Weakness Enumeration - <http://cwe.mitre.org/>
* PCI Security Standards Council - <https://www.pcisecuritystandards.org>
* PCI Data Security Standard (DSS) v3.0 Requirements and Security Assessment Procedures <https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf>