



OWASP

The Open Web Application Security Project

06/09

OWASP Application Security Verification Standard 2009

– Web Application Standard

release



Creative Commons (CC) Attribution Share-Alike
Free version at <http://www.owasp.org>



Foreword

This document defines four levels of application-level security verification for Web applications. 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. Each level described in this document includes a set of requirements for verifying the effectiveness of security controls that protect Web applications.

The requirements were developed with the following objectives in mind:

- *Use as a metric* - Provide application developers and application owners with a yardstick with which to assess the degree of trust that can be placed in their Web applications,
- *Use as guidance* - Provide guidance to security control developers as to what to build into security controls in order to satisfy application security requirements,¹ and
- *Use during procurement* - Provide a basis for specifying application security verification requirements in contracts.²

The requirements were designed to meet the above objectives by ensuring validation of how security controls are designed, implemented, and used by an application. The requirements ensure that the security controls used by an application operate using a deny-by-default strategy, are centralized, are located on the server side, and are all used where necessary.

Copyright and License

Copyright © 2008 - 2009 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.

¹ For more information about building and using security controls that meet ASVS requirements, see the *Enterprise Security API (ESAPI)* (OWASP, 2009).

² For more information about using ASVS in contracts, see the *Contract Annex* (OWASP, 2009).



Table of Contents

Introduction.....	1
Approach	1
Acknowledgements	3
Application Security Verification Levels	4
Level 1 - Automated Verification	4
Level 1A - Dynamic Scan (Partial Automated Verification)	6
Level 1B - Source Code Scan (Partial Automated Verification).....	7
Level 2 - Manual Verification	7
Level 2A - Security Test (Partial Manual Verification)	10
Level 2B - Code Review (Partial Manual Verification).....	10
Level 3 - Design Verification.....	10
Level 4 - Internal Verification	13
Requirement Interpretations and Precedents	15
Detailed Verification Requirements	16
V1 - Security Architecture Documentation Requirements.....	17
V2 - Authentication Verification Requirements	18
V3 - Session Management Verification Requirements	19
V4 - Access Control Verification Requirements	20
V5 - Input Validation Verification Requirements.....	22
V6 - Output Encoding/Escaping Verification Requirements.....	23
V7 - Cryptography Verification Requirements.....	24
V8 - Error Handling and Logging Verification Requirements	25
V9 - Data Protection Verification Requirements	26
V10 - Communication Security Verification Requirements	27
V11 - HTTP Security Verification Requirements.....	28
V12 - Security Configuration Verification Requirements	29
V13 - Malicious Code Search Verification Requirements.....	30
V14 - Internal Security Verification Requirements	31
Verification Reporting Requirements.....	32
R1 - Report Introduction	32
R2 - Application Description	32
R3 - Application Security Architecture	32
R4 - Verification Results.....	33
Glossary	35
Where To Go From Here.....	37



Figures

Figure 1 - OWASP ASVS Levels	1
Figure 2 - One way to introduce verification as an activity into your SDLC	2
Figure 3 - OWASP ASVS Levels 1, 1A, and 1B	5
Figure 4 - OWASP ASVS Level 1 Security Architecture Example	6
Figure 5 - OWASP ASVS Levels 2, 2A, and 2B	7
Figure 6 - OWASP ASVS Level 2 Security Architecture Example	9
Figure 7 - OWASP ASVS Level 3	11
Figure 8 - OWASP ASVS Level 3 Security Architecture Example	12
Figure 9 - OWASP ASVS Level 4	13
Figure 10 - OWASP ASVS Level 4 Unexamined Code Example	15
Figure 11 - Report Contents	32

Tables

Table 1 - OWASP ASVS Security Architecture Requirements (V1)	17
Table 2 - OWASP ASVS Authentication Requirements (V2)	18
Table 3 - OWASP ASVS Session Management Requirements (V3)	19
Table 4 - OWASP ASVS Access Control Requirements (V4)	21
Table 5 - OWASP ASVS Input Validation Requirements (V5)	22
Table 6 - OWASP ASVS Output Encoding/Escaping Requirements (V6)	23
Table 7 - OWASP ASVS Cryptography Requirements (V7)	24
Table 8 - OWASP ASVS Error Handling and Logging Requirements (V8)	25
Table 9 - OWASP ASVS Data Protection Requirements (V9)	26
Table 10 - OWASP ASVS Communication Security Requirements (V10)	27
Table 11 - OWASP ASVS HTTP Security Requirements (V11)	29
Table 12 - OWASP ASVS Security Configuration Requirements (V12)	29
Table 13 - OWASP ASVS Malicious Code Search Requirements (V13)	30
Table 14 - OWASP ASVS Internal Security Requirements (V14)	31
Table 15 - OWASP ASVS Report Verification Results Contents	33



Introduction

The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security. We advocate approaching application security as a people, process, and technology problem, because the most effective approaches to application security include improvements in all of these areas. We can be found at www.owasp.org.

OWASP is a new kind of organization. Our freedom from commercial pressures allows us to provide unbiased, practical, cost-effective information about application security. OWASP is not affiliated with any technology company, although we support the informed use of commercial security technology. Similar to many open-source software projects, OWASP produces many types of materials in a collaborative, open way. The OWASP Foundation is a not-for-profit entity that ensures the project's long-term success.

The primary aim of the OWASP Application Security Verification Standard (ASVS) Project is to normalize the range in the coverage and level of rigor available in the market when it comes to performing Web application security verification using a commercially-workable open standard. The standard provides a basis for testing application technical security controls, as well as any technical security controls in the environment, that are relied on to protect against vulnerabilities such as Cross-Site Scripting (XSS) and SQL injection.³ This standard can be used to establish a level of confidence in the security of Web applications.

Approach

The OWASP ASVS defines verification and documentation requirements that are grouped on the basis of related coverage and level of rigor. The Standard defines four hierarchical levels (e.g. Level 2 requires more coverage and rigor than Level 1) as depicted in the figure below.

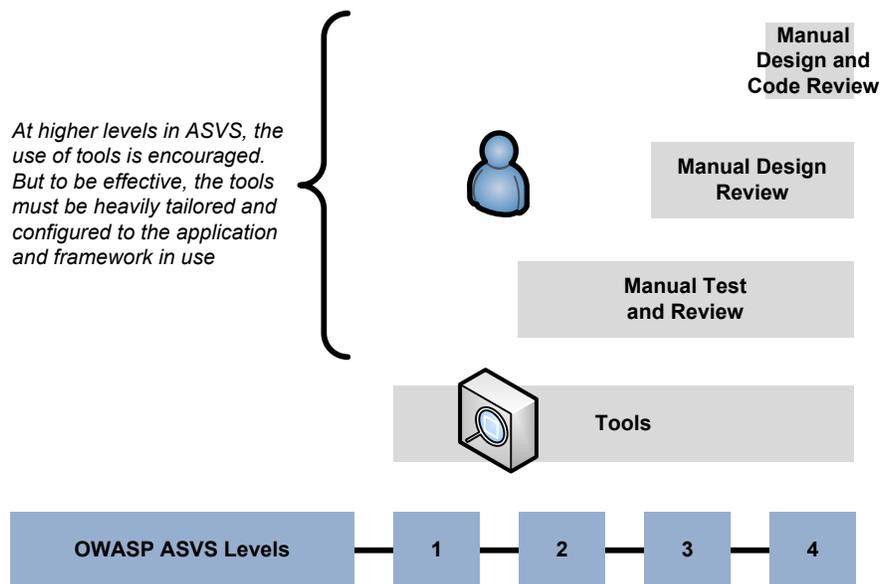


Figure 1 - OWASP ASVS Levels

³ For more information about common Web application vulnerabilities, see the *OWASP Top Ten* (OWASP, 2007).



Web application security verification is performed from a logical point of view by following (or attempting to follow) paths into and out of a targeted application (called the Target of Verification or TOV) and performing analysis along those paths. More complex applications typically take more time to analyze resulting in longer and more costly verifications. Lines of code are not the only factors that determine the complexity of an application - different technologies will typically require different amounts of analysis. Simple applications may include for example libraries and frameworks. Applications of moderate complexity may include simple Web 1.0 applications. Complex applications may include Web 2.0 applications and new/unique Web technologies.

ASVS defines constituent components for Levels 1 and 2 (e.g. verification at Level 1 requires meeting both Level 1A and 1B requirements). For example, applications may claim compliance to either Level 1A or 1B instead of Level 1, but making such claims is weaker than claiming Level 1. Verification and documentation requirements are defined in this Standard using three types of requirements: High-Level requirements, Detailed requirements, and Reporting requirements. The High-Level requirements define the overall application implementation and verification requirements. The Detailed requirements define low-level application implementation and verification requirements (i.e., specific items to verify). The Reporting requirements define how the results of performing an application verification according to the OWASP ASVS must be documented.

OWASP provides numerous resources, including ASVS, to help organization's develop and maintain secure applications. The OWASP ASVS, OWASP Contract Annex,⁴ and OWASP ESAPI⁵ can be used to support your Software Development Life Cycle (SDLC) as depicted in the figure below.

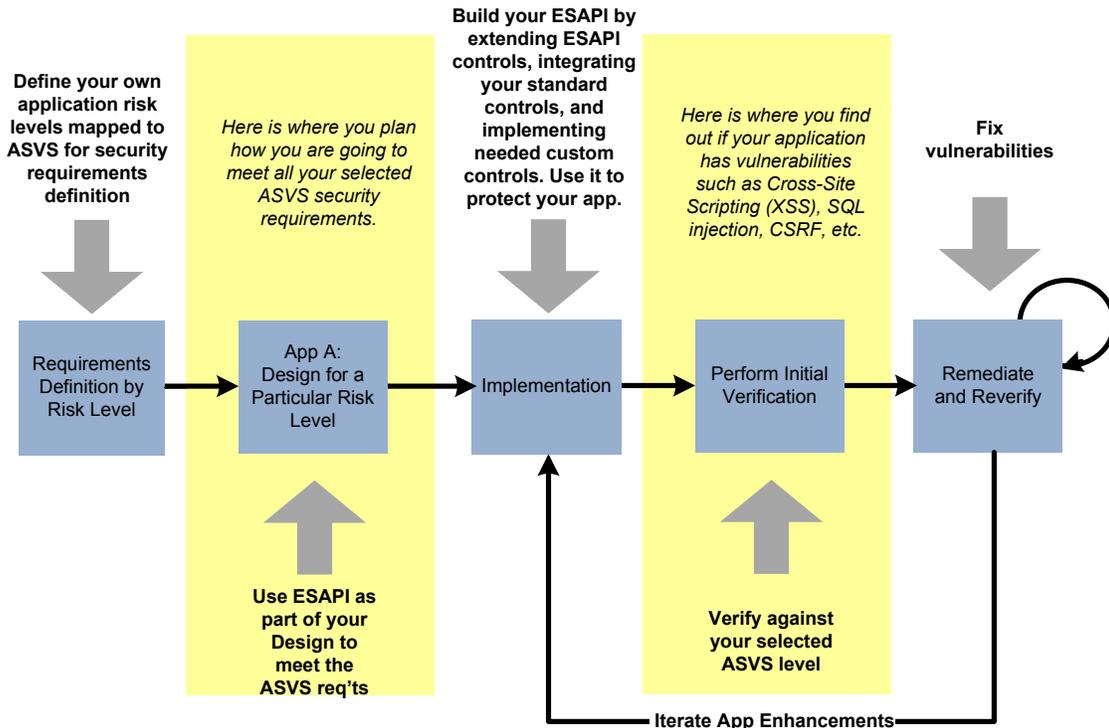


Figure 2 - One way to introduce verification as an activity into your SDLC⁶

⁴ For information about how to specify an ASVS level in a contract, see the *OWASP Contract Annex*.

⁵ For more information about how to ESAPI-Enable (ES-Enable) your application, see the *OWASP ESAPI* project (OWASP 2009).

⁶ For more information about introducing security-related activities into your existing SDLC, see the *OWASP CLASP* (OWASP 2008) or *OWASP SAMM* Projects (OWASP 2009).



Acknowledgements

We thank the OWASP Foundation for sponsoring the OWASP Application Security Verification Standard Project during the OWASP Summer of Code 2008.

Project Lead: ⁷ Mike Boberski (Booz Allen Hamilton)

Authors: ⁸ Mike Boberski (Booz Allen Hamilton), Jeff Williams (Aspect Security), Dave Wichers (Aspect Security)

Project
Sponsors:



Booz | Allen | Hamilton

Acknowledgement is given for the contributions of: Pierre Parrend, who acted as an OWASP Summer of Code 2008 Reviewer; Andrew van der Stock (Aspect Security); Nam Nguyen (Blue Moon Consulting); John Martin (Boeing); Gaurang Shah (Booz Allen Hamilton); Theodore Winograd (Booz Allen Hamilton); Stan Wisseman (Booz Allen Hamilton); Barry Boyd (CGI Federal); Steve Coyle (CGI Federal); Paul Douthit (CGI Federal); Ken Huang (CGI Federal); Dave Hausladen (CGI Federal); Mandeep Khara (Cenzic); Scott Matsumoto (Cigital); John Steven (Cigital); Stephen de Vries (Corsaire); Dan Cornell (Denim Group); Shouvik Bardhan (Electrosoft), Dr. Sarbari Gupta (Electrosoft); Eoin Keary (Ernst & Young); Richard Campbell (Federal Deposit Insurance Corporation); Matt Presson (FedEx); Jeff LoSapio (Fortify Software); Liz Fong (National Institute of Standards and Technology); George Lawless (Noblis); Dave van Stein (ps_testware); Terrie Diaz (SAIC); Ketan Dilipkumar Vyas (Tata Consultancy Services); Bedirhan Urgan (TURKCELL); Dr. Thomas Braun (United Nations); Colin Watson (Watson Hall); Jeremiah Grossman (WhiteHat Security); and finally, thanks are given to the application security verification community and others interested in trusted Web computing for their enthusiastic advice and assistance throughout this effort.

⁷ Email: mike.boberski@owasp.org

⁸ Email: jeff.williams@owasp.org, dave.wichers@owasp.org



Application Security Verification Levels

The ASVS defines four levels of verification that increase in both breadth and depth as one moves up the levels. The breadth is defined in each level by a set of security requirements that must be addressed. The depth of the verification is defined by the approach and level of rigor required in verifying each security requirement. Tools are an important part of every ASVS level. At higher levels in ASVS, the use of tools is encouraged. But to be effective, the tools must be heavily tailored and configured to the application and framework in use. And, at all levels, tool results must be manually verified.

It is a verifier's responsibility to determine if a TOV meets all of the requirements at the level targeted by a review. If the application meets all of the requirements for that level, then it can be considered an OWASP ASVS Level N application, where N is the verification level that application complied with. If the application does not meet all the requirements for a particular level, but does meet all the requirements for a lower level of this standard, then it can be considered to have passed the lower level of verification. This standard uses the term the 'verifier' to indicate the person or team that is reviewing the application against these requirements.

The specification for an application may require OWASP ASVS Level N, but it could also include other additional detailed requirements such as from a higher ASVS level. For example, a financial organization may have a lower-risk application verified to OWASP ASVS Level 2 but may also want verification that no malicious code (see V13, Level 4 only) has been included. Other organization or business requirements could apply, such as compliance with particular information security policies and regulations.



There is no verification level 0. Also, to earn a level, vulnerabilities must be remediated (or mitigated), and the application re-verified.

Level 1 - Automated Verification

Level 1 ("Automated Verification") is typically appropriate for applications where some confidence in the correct use of security controls is required. Threats to security⁹ will typically be viruses and worms (targets are chosen indiscriminately through wide scans and impact the most vulnerable). The scope of verification includes code that was developed or modified in order to create the application.

In Level 1, the verification involves the use of automated tools augmented with manual verification. This level only provides partial application security verification coverage. The manual verification is not intended to make the application security verification performed at this level complete, only to verify that each automated finding is correct and not a false positive.

There are two constituent components for Level 1. Level 1A is for the use of automated application vulnerability scanning (dynamic analysis) tools, and Level 1B is for the use of automated source code scanning (static analysis) tools. Verification efforts may use either of these components individually, or may perform a combination of these approaches to achieve a complete Level 1 rating. The structure of these levels is depicted in the figure below.

While it may be determined that an application meets either Level 1A or 1B, neither of these levels alone provide the same levels of rigor or coverage as an application that meets Level 1. An application that meets Level 1 must meet both Level 1A and 1B requirements.

⁹ For more information about identifying risks and estimating risks associated with vulnerabilities, see the *OWASP Testing Guide* (OWASP, 2008).

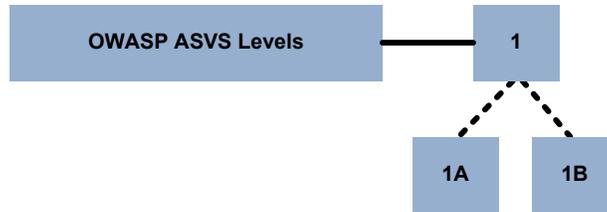


Figure 3 - OWASP ASVS Levels 1, 1A, and 1B

The following are the minimal high-level requirements for Level 1, 1A, or 1B applications:

Verification Scope

L1.1 The scope of the verification includes all code that was developed or modified in order to create the application.

Security Control Behavior Requirements

None There are no requirements for how application security controls make decisions at Level 1.

Security Control Use Requirements

None There are no requirements for where application security controls are used within the application at Level 1.

Security Control Implementation Requirements

None There are no requirements for how the application security controls are built at Level 1.

Security Control Verification Requirements

L1.2 Dynamically scan the application according to the Level 1A requirements in the “Detailed Verification Requirements” section.

L1.3 Perform source code scanning on the application according to the Level 1B requirements in the “Detailed Verification Requirements” section.

Requirements at Level 1 that allow the use of either verification technique only have to be verified with one technique. In addition, if the verifier’s selected tool suite does not have the capability to verify a specified verification requirement, the verifier can perform manual verification to fill this gap.^{10 11}

¹⁰ For more information about performing manual verification by performing manual penetration testing, see the *OWASP Testing Guide* (OWASP, 2008).

¹¹ For more information about performing manual verification by performing a manual code review, see the *OWASP Code Review Guide* (OWASP, 2008).



Reporting Requirements

L1.4 Create a verification report that details the application’s security architecture by listing its components, and includes the results of the verification according to the requirements in the “Verification Reporting Requirements” section.

At Level 1, application components may be defined in terms of either individual or groups of source files, libraries, and/or executables, as depicted in the figure below. At Level 1, the list need not be sorted or otherwise organized other than identifying which components are part of the application, and which components are part of the IT environment. The application can then be treated as groups of components within a single monolithic entity. The path or paths a given end user request may take within the application do not need to be identified and documented.

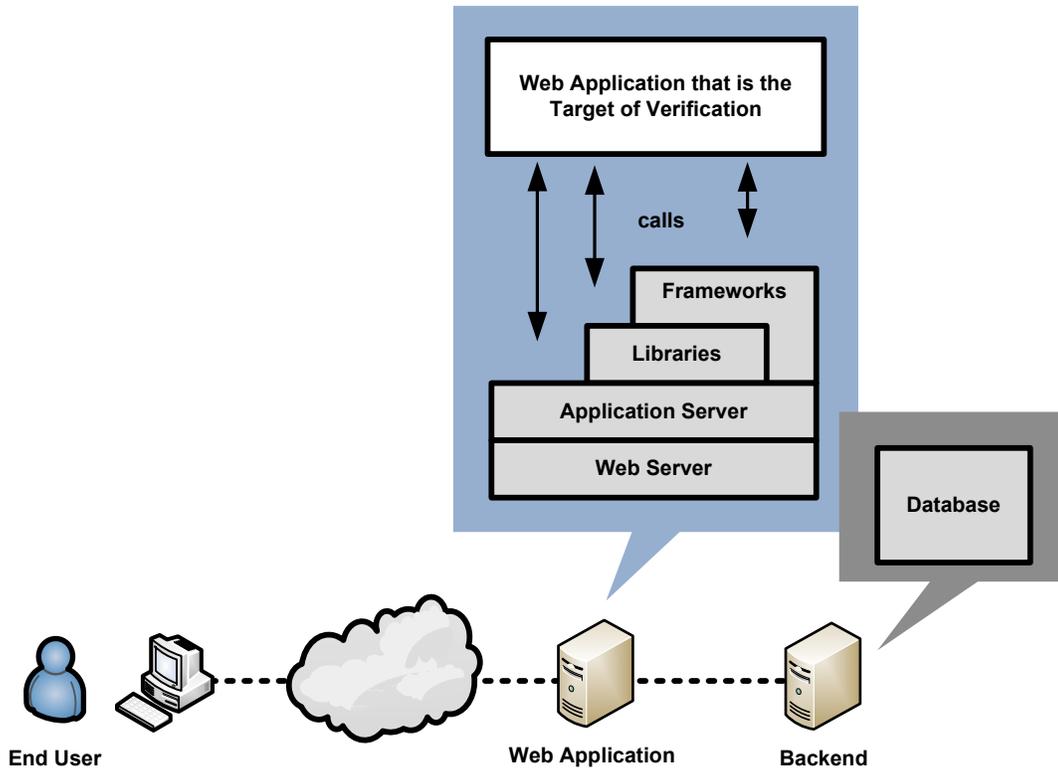


Figure 4 - OWASP ASVS Level 1 Security Architecture Example

Level 1A - Dynamic Scan (Partial Automated Verification)

Dynamic Scanning Security Control Verification Requirements

Dynamic scanning (also known as “application vulnerability scanning”) consists of using automated tools to access application interfaces, while the application is running, in order to detect vulnerabilities in the application’s security controls. Note that this is not sufficient to verify the correct design, implementation, and use of a security control, but is acceptable verification at Level 1. The scope of verification is defined by this Level’s security architecture requirements.

L1A.1 Dynamically scan the application according to the Level 1A requirements specified in the “Detailed Verification Requirements” section.



L1A.2 Verify all dynamic scan results using either manual application penetration testing or code review. Unverified automated results are not considered to provide any assurance and are not sufficient to qualify for Level 1.

Multiple instances of a particular type of vulnerability that can be traced to a single root cause should be combined into a single finding if the scanning tool does not already do so.

Level 1B - Source Code Scan (Partial Automated Verification)

Source Code Scanning Security Control Verification Requirements

Source code scanning (also known as “static analysis”) consists of using automated tools to search through the application source code to find patterns that represent vulnerabilities. Note that this is not sufficient to verify the correct design, implementation, and use of a security control, but is acceptable verification at Level 1. The scope of verification is defined by this Level’s security architecture requirements.

L1B.1 Perform source code scanning on the application according to the Level 1B requirements specified in the “Detailed Verification Requirements” section.

L1B.2 Verify all source code scan results using either manual application penetration testing or code review. Unverified automated results are not considered to provide any assurance and are not sufficient to qualify for Level 1.

Multiple instances of a particular type of vulnerability that can be traced to a single root cause should be combined into a single finding if the code analysis tool does not already do so.

Level 2 - Manual Verification

Level 2 (“Manual Verification”) is typically appropriate for applications that handle personal transactions, conduct business-to-business transactions, process credit card information, or process personally identifiable information. Level 2 provides some confidence in the correct use of security controls and confidence that the security controls are working correctly. Threats to security will typically be viruses, worms, and unsophisticated opportunists such as attackers with professional or open source attack tools. The scope of verification includes all code developed or modified for the application as well as examining the security of all third party components that provide security functionality for the application. There are two constituent components for Level 2, as depicted in the figure below.

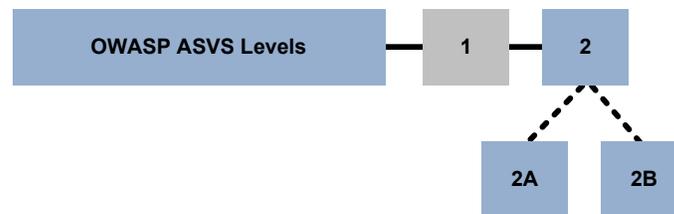


Figure 5 - OWASP ASVS Levels 2, 2A, and 2B

While it may be determined that an application meets either Level 2A or 2B, neither of these levels alone provide the same levels of rigor or coverage as Level 2. Further, while Level 2 is a superset of Level 1, there is no requirement to run an automated tool to meet the Level 2 requirements. Instead, the verifier has the option of using just manual techniques for all requirements. If automated tool results are available, the verifier may use them to support the analysis. However, even passing a requirement at Level 1 does not automatically indicate passing the same requirement



at Level 2. This is because automated tools do not provide sufficient evidence that the positive requirement has been met.

Manual techniques are still assumed to employ the use of tools. This can include the use of any kind of security analysis or testing tool, including the automated tools that are used for Level 1 verifications. However, such tools are simply aids to the analyst to find and assess the security controls being verified. Such tools may or may not contain logic to automatically detect application vulnerabilities.

The following are the minimal high-level requirements for Level 2, 2A, or 2B applications:

Verification Scope

- | | |
|------|--|
| L2.1 | The scope of the verification includes all code that was developed or modified in order to create the application. This requirement was introduced at Level 1. |
| L2.2 | The scope of the verification includes the code for all third-party framework, library, and service security functionality that is invoked by or supports the security of the application. This is a new requirement at Level 2. |

Security Control Behavior Requirements

- | | |
|------|--|
| L2.3 | Verify that all technical security controls that perform security checks make decisions using a whitelist approach. This is a new requirement at Level 2. |
| L2.4 | Verify that all security controls that perform security checks and security controls that result in security effects cannot be bypassed according to the Level 2A and 2B requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 2. |

Security Control Use Requirements

- | | |
|------|--|
| L2.5 | Verify that all security controls are used everywhere within the application they need to be, and the implementations are centralized within the application, on the server side, according to the Level 2 requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 2. |
|------|--|

Security Control Implementation Requirements

- | | |
|------|---|
| None | There are no requirements for how application security controls are built at Level 2. |
|------|---|

Security Control Verification Requirements

- | | |
|------|---|
| L2.6 | Perform manual application penetration testing on the application according to the Level 2A requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 2. |
| L2.7 | Perform a manual source code review on the application according to the Level 2B requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 2. |

Requirements at Level 2 that allow the use of either verification technique only have to be verified with one technique.

The verifier may include automated scanning or code analysis as part of their verification effort at Level 2, but automated verification cannot be used in place of the manual review required for each Level 2 requirement. If the scan results help the verifier perform their work more quickly or



augment the results of the manual portion of the review, they can certainly be used to assist in performing a Level 2 verification.

Reporting Requirements

L2.8 Create a verification report that describes the application’s security architecture by grouping its components into a high-level architecture, and includes the results of the verification according to the requirements in the “Verification Reporting Requirements” section. This augments the reporting requirement introduced at Level 1.

At Level 2, application components may be defined in terms of either individual or groups of source files, libraries, and/or executables that are organized into a high-level architecture (for example Model-View-Controller (MVC) components, business function components, and data layer components). For example, the diagram below depicts an application that consists of a server application, an application server application, custom code, libraries, and a database application that are grouped according to an MVC architecture. At Level 2, the path or paths a given *end user* request may take within the application must be documented, as depicted in the figure below. However, not all such paths must be examined.

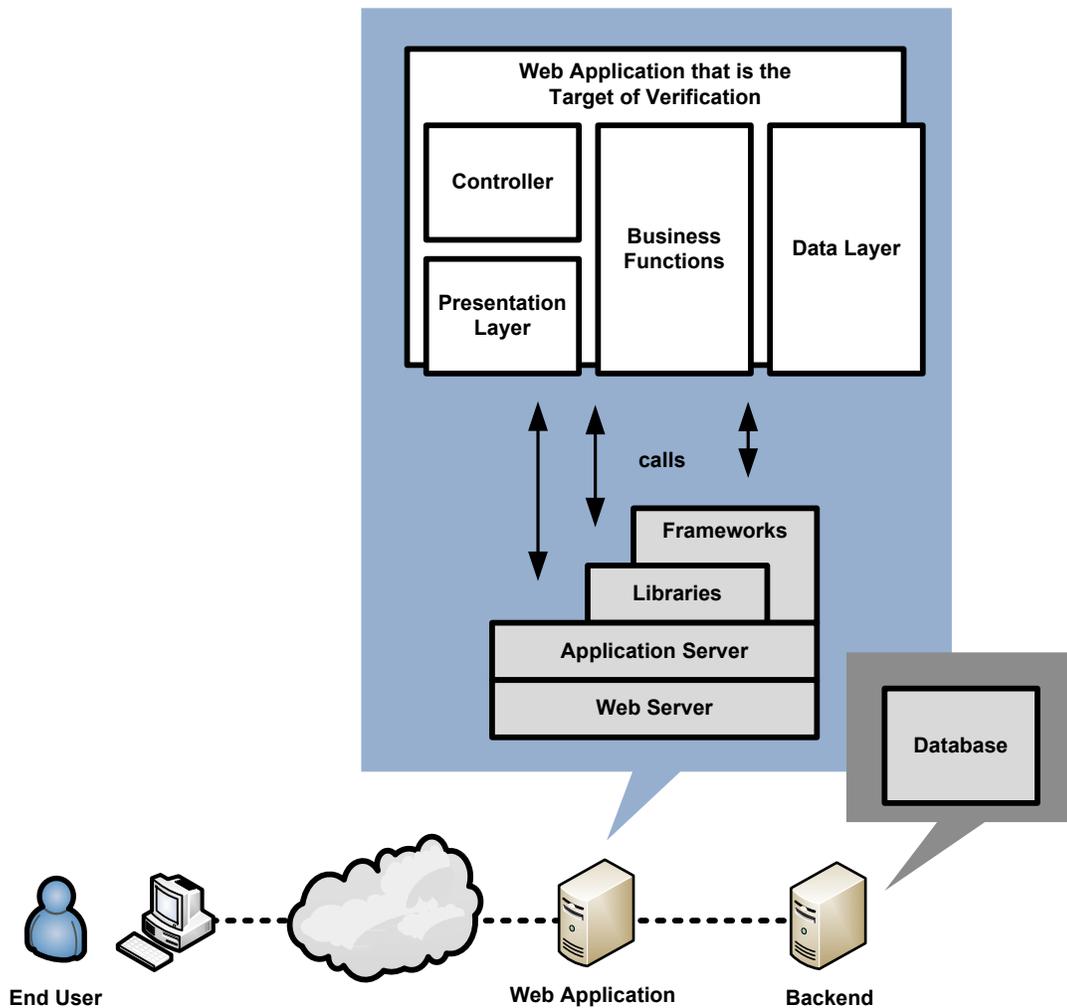


Figure 6 - OWASP ASVS Level 2 Security Architecture Example



Level 2A - Security Test (Partial Manual Verification)

Manual Application Penetration Testing Security Control Verification Requirements

Manual application security testing consists of creating dynamic tests to verify an application's proper design, implementation, and use of security controls. The scope of verification is defined by this Level's security architecture requirements.

L2A.1 Perform manual application security testing on the application according to the Level 2A requirements specified in the "Detailed Verification Requirements" section. This is a new requirement at Level 2.

Where appropriate, the verifier may use sampling to establish the effective use of a security control. The verifier may choose to document a vulnerability pattern that will allow developers to confidently find and fix all instances of the pattern in the software baseline. Multiple instances of a vulnerability pattern that can be traced to a single root cause should be combined into a single finding.

Level 2B - Code Review (Partial Manual Verification)

Manual Code Review Security Control Verification Requirements

Manual code review consists of human searching and analysis of the application source code in order to verify the application's design, implementation, and proper use of security controls. Such analysis is expected to be tool assisted, but could simply involve commonly available tools such as a source code editor or IDE. The scope of verification is defined by this Level's security architecture requirements.

L2B.1 Perform a manual code review on the application according to the Level 2B requirements specified in the "Detailed Verification Requirements" section. This is a new requirement at Level 2.

Where appropriate, the verifier may use an appropriate sampling method to establish the effective use of a security control. The verifier may choose to document a vulnerability pattern that will allow developers to confidently find and fix all instances of the vulnerability pattern in the software baseline. Multiple instances of a vulnerability pattern that can be traced to a single root cause should be combined into a single finding.

Level 3 - Design Verification

Level 3 ("Design Verification") 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 security will typically be viruses and worms, opportunists, and possibly determined attackers (skilled and motivated attackers focusing on specific targets using tools including purpose-built scanning tools). The scope of verification includes all code developed or modified for the application, as well as examining the security of all third party components that provide security functionality for the application. Level 3 ensures that security controls themselves are working correctly, and that security controls are used everywhere within the application they need to be used to enforce application-specific policies. Level 3 is not broken into multiple components, as depicted in the figure below.

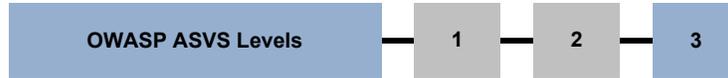


Figure 7 - OWASP ASVS Level 3

The following are the minimal high-level requirements for Level 3 applications:

Verification Scope

- L3.1 The scope of the verification includes all code that was developed or modified in order to create the application. This requirement was introduced at Level 1.
- L3.2 The scope of the verification includes the code for all third-party framework, library, and service security functionality that is invoked by or supports the security of the application. This requirement was introduced at Level 2.
- L3.3 The scope of the verification includes the code for all third-party frameworks, libraries, and services associated with the application. This is a new requirement at Level 3.

Security Control Behavior Requirements

- L3.4 Verify that all security controls that perform security checks make decisions using a whitelist approach. This requirement was introduced at Level 2.
- L3.5 Verify that all security controls that perform security checks and security controls that result in security effects cannot be bypassed according to the Level 3 requirements specified in the “Detailed Verification Requirements” section. This requirement was introduced at Level 2.

Security Control Use Requirements

- L3.6 Verify that all security controls are used everywhere within the application they need to be, and the implementations are centralized within the application, on the server side, according to the Level 3 requirements specified in the “Detailed Verification Requirements” section. This requirement was introduced at Level 2.

Security Control Implementation Requirements

- None There are no requirements for how application security controls are built at Level 3.

Security Control Verification Requirements

- L3.7 Manually verify the application according to the Level 3 requirements specified in the “Detailed Verification Requirements” section. This augments the manual verification requirements introduced at Level 2.
- L3.8 Document a security architecture and use it to verify the proper design and use of all security controls by performing threat modeling. This is a new requirement at Level 3.

Reporting Requirements

- L3.9 Create a verification report that describes the application’s security architecture by grouping its components into a high-level architecture that includes threat modeling information, and includes the results of the verification according to the requirements in



the “Verification Reporting Requirements” section. This augments the reporting requirement from Level 2.

At level 3, application components may be defined in terms of either individual or groups of source files, libraries, and/or executables that are grouped into a high-level architecture (for example MVC components, business function components, and data layer components). At Level 3, supporting threat modeling information about threat agents and assets must additionally be provided. The path or paths a given end user request may take through a high-level view of the application must be documented, as depicted in the figure below. At Level 3, *all* potential paths through the high-level view of the application must be examined.

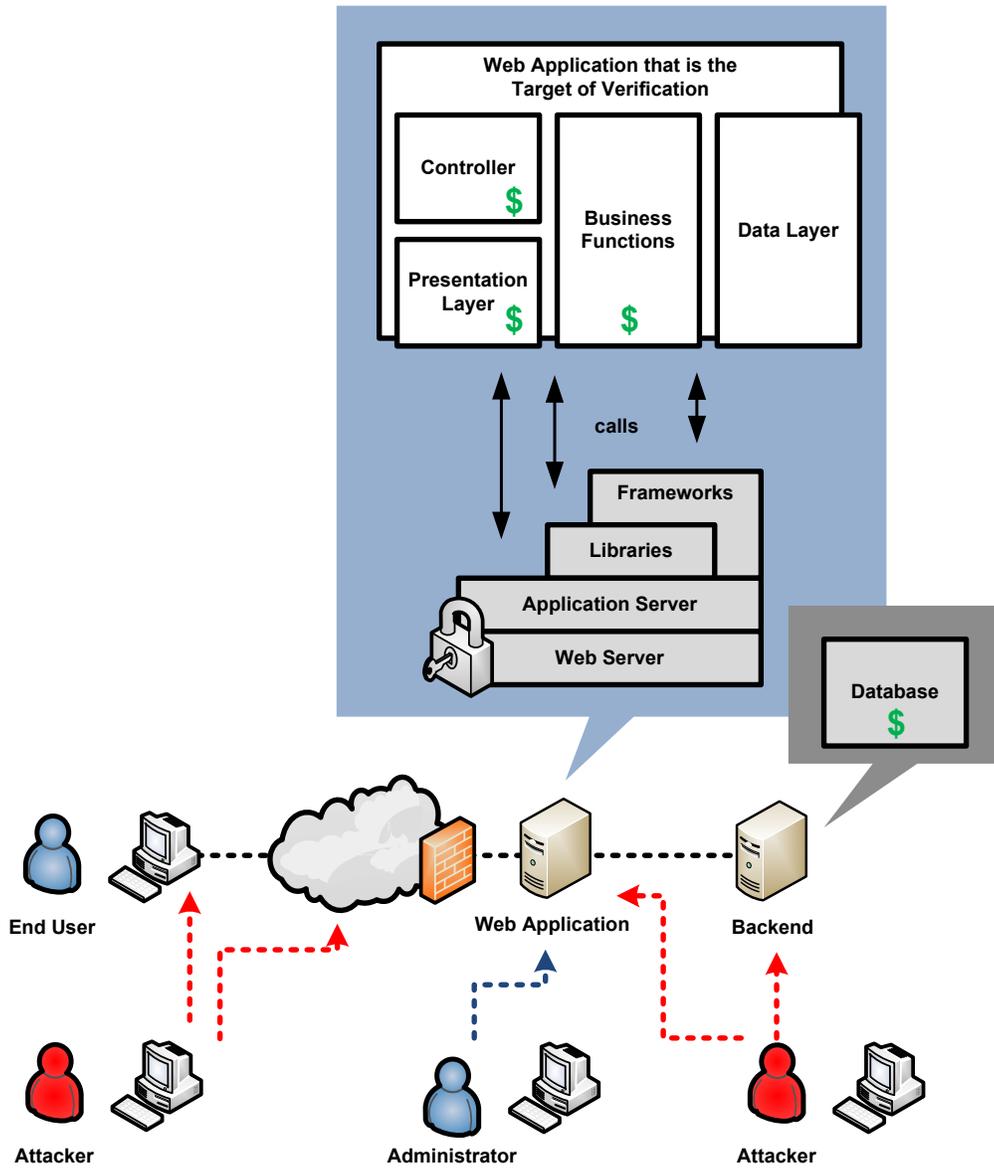


Figure 8 - OWASP ASVS Level 3 Security Architecture Example¹²

¹² Dollar signs indicate assets in the diagram.



Level 4 - Internal Verification

Level 4 (“Internal Verification”) is typically appropriate for critical applications that protect life and safety, critical infrastructure, or defense functions. Level 4 may also be appropriate for applications that process sensitive assets. Level 4 ensures that security controls themselves are working correctly, that security controls are used everywhere within the application they need to be used to enforce application-specific policies, and that secure coding practices were followed. Threats to security will be from determined attackers (skilled and motivated attackers focusing on specific targets using tools including purpose-built scanning tools). The scope of verification expands beyond the scope of Level 3 to include all code used by the application. Level 4 is not broken out into constituent components, as depicted in the figure below.



Figure 9 - OWASP ASVS Level 4

The following are the minimal high-level requirements for Level 4 applications:

Verification Scope

- L4.1 The scope of the verification includes all code that was developed or modified in order to create the application. This requirement was introduced at Level 1.
- L4.2 The scope of the verification includes the code for all third-party framework, library, and service security functionality that is invoked by or supports the security of the application. This requirement was introduced at Level 2.
- L4.3 The scope of the verification includes the code for all third-party frameworks, libraries, and services associated with the application. This requirement was introduced at Level 3.
- L4.4 The scope of the verification includes all remaining code associated with the application, including frameworks, libraries, runtime environments, development tools, build tools, and deployment tools. The scope does not include the code for platform software, such as an application server, database server, virtual machine, or operating system, that has received a substantial amount of public scrutiny. This is a new requirement at Level 4.

Security Control Behavior Requirements

- L4.5 Verify that all security controls that perform security checks make decisions using a whitelist (“positive”) approach. This requirement was introduced at Level 2.
- L4.6 Verify that all security controls that perform security checks and security controls that result in security effects cannot be bypassed according to the Level 4 requirements specified in the “Detailed Verification Requirements” section. This requirement was introduced at Level 2.

Security Control Use Requirements

- L4.7 Verify that all security controls are used everywhere within the application they need to be, and that the implementations are centralized within the application, on the server side, according to the Level 4 requirements specified in the “Detailed Verification Requirements” section. This requirement was introduced at Level 3.



Security Control Implementation Requirements

- | | |
|------|--|
| L4.8 | Verify that the application does not contain any malicious code according to the Level 4 requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 4. |
|------|--|

Security Control Verification Requirements

- | | |
|-------|---|
| L4.9 | Manually verify the application against the Level 4 requirements specified in the “Detailed Verification Requirements” section. This augments the requirement from Level 3. |
| L4.10 | Document a security architecture and use it to verify the proper design and use of all security controls by performing threat modeling. This requirement was introduced at Level 3. |
| L4.11 | Manually review all code developed or modified for this application for malicious code ¹³ according to the Level 4 requirements specified in the “Detailed Verification Requirements” section. This is a new requirement at Level 4. |

Reporting Requirements

- | | |
|-------|---|
| L4.12 | Create a verification report that describes the application’s security architecture according to the Level 3 requirements, which encompasses all application code, and includes the results of the verification according to the requirements in the “Verification Reporting Requirements” section. This augments the reporting requirement from Level 3. |
|-------|---|

At Level 4, the application’s architecture shall be captured as required at Level 3. Further, Level 4 requires that all application code, including code not explicitly examined, be identified as part of the application definition, as depicted in the figure below. This code must include all libraries, frameworks, and supporting code that the application relies on. Previous verifications of these components can be reused as part of another verification effort. Platform code, such as the operating system, virtual machine, or core libraries issued with a virtual machine environment, Web server, or application server are not included in Level 4. For example, libraries associated with the Java runtime would not need to be assessed at Level 4.

¹³ Malicious code is not the same as malware. See the glossary definition of malicious code.

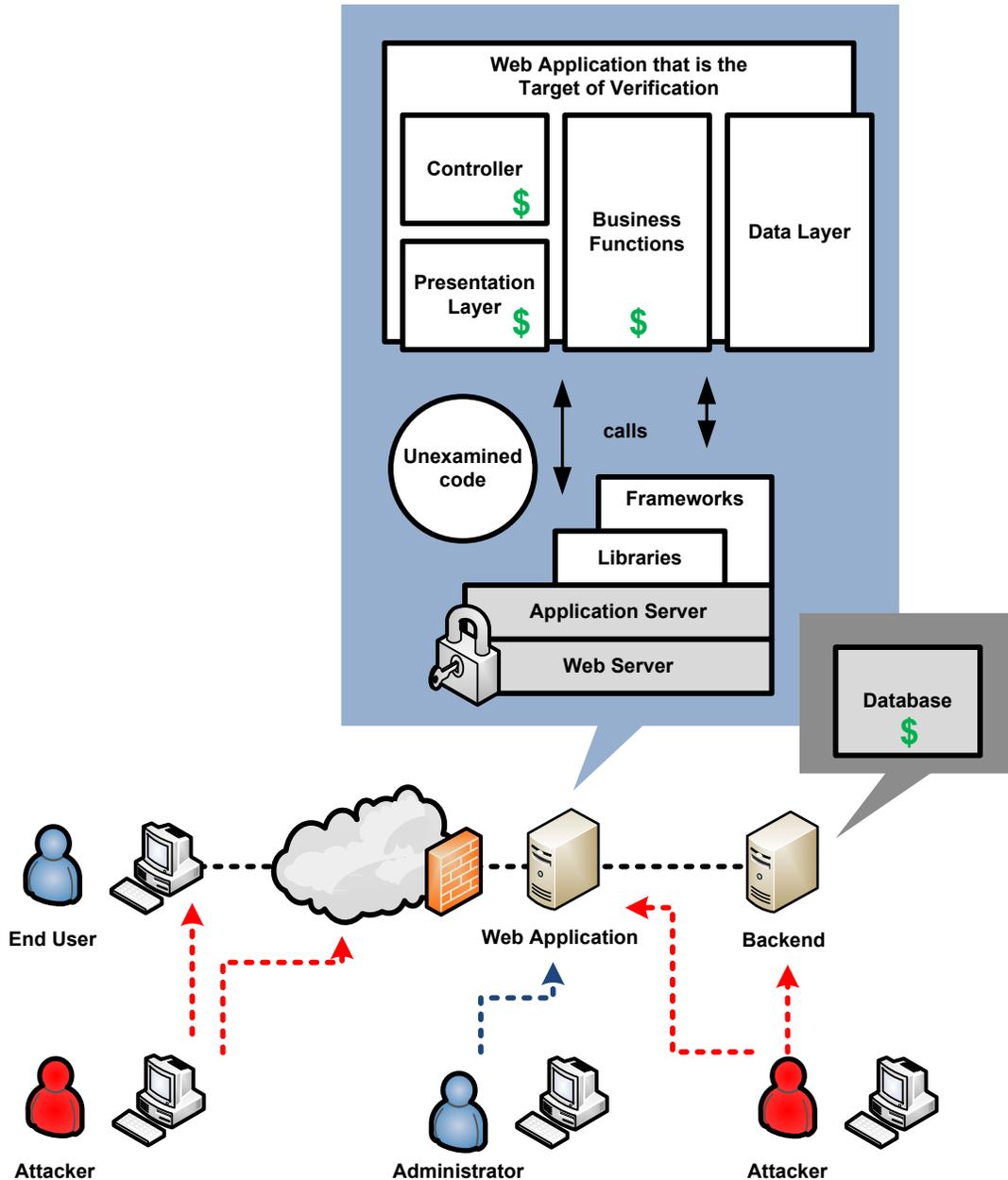


Figure 10 - OWASP ASVS Level 4 Unexamined Code Example

Requirement Interpretations and Precedents

The OWASP ASVS is a living document. If you are performing an application security verification according to this standard, then you should always review the articles that can be found on the OWASP ASVS project page at the following location: http://www.owasp.org/index.php/ASVS#Articles_Below_-_More_About_ASVS_and_Using_It. The articles on the OWASP ASVS project page provide requirement clarifications, requirement verdict precedents, and helpful hints.



Detailed Verification Requirements

This section of the OWASP Application Security Verification Standard (ASVS) defines detailed verification requirements that were derived from the high-level requirements for each of the verification levels defined in this standard. Each section below defines a set of detailed verification requirements grouped into related areas.

The ASVS defines the following security requirements areas:

- V1. Security Architecture
- V2. Authentication
- V3. Session Management
- V4. Access Control
- V5. Input Validation
- V6. Output Encoding/Escaping
- V7. Cryptography
- V8. Error Handling and Logging
- V9. Data Protection
- V10. Communication Security
- V11. HTTP Security
- V12. Security Configuration
- V13. Malicious Code Search
- V14. Internal Security

For each of these areas, the requirements that must be met at each of the verification levels listed below are specified:

- Level 1: Automated Verification
 - Level 1A - Dynamic Scan (Partial Automated Verification)
 - Level 1B - Source Code Scan (Partial Automated Verification)
- Level 2: Manual Verification
 - Level 2A - Security Test (Partial Manual Verification)
 - Level 2B - Code Review (Partial Manual Verification)
- Level 3: Design Verification
- Level 4: Internal Verification



V1 - Security Architecture Documentation Requirements

For all ASVS levels, documenting some basic security architecture information is necessary to ensure both the completeness and accuracy (and repeatability when remediation is required) of the application security verification that is performed. Analysis can be directed and results can be traced back to the application's high level security architecture. These requirements start with a base level of security architecture detail that must be captured and this level of detail grows with each level. The table below defines the corresponding security architecture documentation requirements that apply for each of the four verification levels.

Table 1 - OWASP ASVS Security Architecture Requirements (V1)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V1.1 Verify that all application components (either individual or groups of source files, libraries, and/or executables) that are present in the application are identified.	✓	✓	✓	✓	✓	✓
V1.2 Verify that all components that are not part of the application but that the application relies on to operate are identified.			✓	✓	✓	✓
V1.3 Verify that a high-level architecture for the application has been defined. ¹⁴			✓	✓	✓	✓
V1.4 Verify that all application components are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.5 Verify that all components that are not part of the application but that the application relies on to operate are defined in terms of the business functions and/or security functions they provide.					✓	✓
V1.6 Verify that threat modeling information has been provided.					✓	✓

¹⁴ The verifier may create or document a high-level design if the application developer does not provide one.



V2 - Authentication Verification Requirements

The Authentication Verification Requirements define a set of requirements for generating and handling account credentials safely. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 2 - OWASP ASVS Authentication Requirements (V2)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V2.1 Verify that all pages and resources require authentication except those specifically intended to be public.	✓	✓	✓	✓	✓	✓
V2.2 Verify that all password fields do not echo the user's password when it is entered.	✓	✓	✓	✓	✓	✓
V2.3 Verify that if a maximum number of authentication attempts is exceeded, the account is locked for a period of time long enough to deter brute force attacks.	✓		✓	✓	✓	✓
V2.4 Verify that all authentication controls are enforced on the server side.			✓	✓	✓	✓
V2.5 Verify that all authentication controls fail securely.			✓	✓	✓	✓
V2.6 Verify that the strength of any authentication credentials are sufficient to withstand attacks that are typical of the threats in the deployed environment.			✓	✓	✓	✓
V2.7 Verify that all account management functions are at least as resistant to attack as the primary authentication mechanism.			✓	✓	✓	✓
V2.8 Verify that users can safely change their credentials using a mechanism that is at least as resistant to attack as the primary authentication mechanism.			✓	✓	✓	✓
V2.9 Verify that re-authentication is required before any application-specific sensitive operations are permitted.			✓	✓	✓	✓
V2.10 Verify that after an administratively-configurable period of time, authentication credentials expire.			✓	✓	✓	✓



V2.11	Verify that all authentication controls (including libraries that call external authentication services) have a centralized implementation.				✓	✓	✓
V2.12	Verify that all authentication decisions are logged.				✓	✓	✓
V2.13	Verify that account passwords are salted using a salt that is unique to that account (e.g., internal user ID, account creation) and hashed before storing.				✓	✓	✓
V2.14	Verify that all authentication credentials for accessing services external to the application are encrypted and stored in a protected location (not in source code).				✓	✓	✓
V2.15	Verify that all code implementing or using authentication controls is not affected by any malicious code.						✓

V3 - Session Management Verification Requirements

The Session Management Verification Requirements define a set of requirements for safely using HTTP requests, responses, sessions, cookies, headers, and logging to manage sessions properly. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 3 - OWASP ASVS Session Management Requirements (V3)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V3.1	Verify that the framework's default session management control implementation is used by the application.		✓	✓	✓	✓
V3.2	Verify that sessions are invalidated when the user logs out.		✓	✓	✓	✓
V3.3	Verify that sessions timeout after a specified period of inactivity.		✓	✓	✓	✓
V3.4	Verify that all pages that require authentication to access them have logout links.		✓	✓	✓	✓



V3.5	Verify that the session id is never disclosed other than in cookie headers; particularly in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.		✓		✓	✓	✓
V3.6	Verify that the session id is changed on login.			✓	✓	✓	✓
V3.7	Verify that the session id is changed on reauthentication.			✓	✓	✓	✓
V3.8	Verify that the session id is changed or cleared on logout.			✓	✓	✓	✓
V3.9	Verify that only session ids generated by the application framework are recognized as valid by the application.			✓		✓	✓
V3.10	Verify that sessions timeout after an administratively-configurable maximum time period regardless of activity (an absolute timeout).					✓	✓
V3.11	Verify that authenticated session tokens are sufficiently long and random to withstand attacks that are typical of the threats in the deployed environment.					✓	✓
V3.12	Verify that cookies which contain authenticated session tokens/ids have their domain and path set to an appropriately restrictive value for that site.					✓	✓
V3.13	Verify that all code implementing or using session management controls is not affected by any malicious code.						✓

V4 - Access Control Verification Requirements

The Access Control Verification Requirements define how an application can safely enforce access control. In most applications, access control must be performed in multiple different locations across the various application layers. These requirements define verification requirements for access controls for URLs, business functions, data, services, and files. The table below defines the corresponding verification requirements that apply for each of the four verification levels.



Table 4 - OWASP ASVS Access Control Requirements (V4)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V4.1 Verify that users can only access protected functions for which they possess specific authorization.	✓	✓	✓	✓	✓	✓
V4.2 Verify that users can only access URLs for which they possess specific authorization.	✓		✓	✓	✓	✓
V4.3 Verify that users can only access data files for which they possess specific authorization.	✓		✓	✓	✓	✓
V4.4 Verify that direct object references are protected, such that only authorized objects are accessible to each user.	✓		✓	✓	✓	✓
V4.5 Verify that directory browsing is disabled unless deliberately desired.	✓		✓		✓	✓
V4.6 Verify that users can only access services for which they possess specific authorization.			✓	✓	✓	✓
V4.7 Verify that users can only access data for which they possess specific authorization.			✓	✓	✓	✓
V4.8 Verify that access controls fail securely.			✓	✓	✓	✓
V4.9 Verify that the same access control rules implied by the presentation layer are enforced on the server side.			✓	✓	✓	✓
V4.10 Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.			✓	✓	✓	✓
V4.11 Verify that all access controls are enforced on the server side.			✓	✓	✓	✓
V4.12 Verify that limitations on input and access imposed by the business on the application (such as daily transaction limits or sequencing of tasks) cannot be bypassed.			✓	✓	✓	✓
V4.13 Verify that there is a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.				✓	✓	✓



V4.14	Verify that there is a centralized mechanism (including libraries that call external authorization services) for protecting access to each type of protected resource.				✓	✓	✓
V4.15	Verify that all access control decisions can be logged and all failed decisions are logged.				✓	✓	✓
V4.16	Verify that all code implementing or using access controls is not affected by any malicious code.						✓

V5 - Input Validation Verification Requirements

The Input Validation Requirements define a set of requirements for validating input so that it is safe for use within an application. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 5 - OWASP ASVS Input Validation Requirements (V5)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V5.1	Verify that the runtime environment is not susceptible to buffer overflows, or that security controls prevent buffer overflows.	✓	✓	✓	✓	✓
V5.2	Verify that a positive validation pattern is defined and applied to all input.	✓	✓	✓	✓	✓
V5.3	Verify that all input validation failures result in input rejection or input sanitization.	✓		✓	✓	✓
V5.4	Verify that a character set, such as UTF-8, is specified for all sources of input.			✓	✓	✓
V5.5	Verify that all input validation is performed on the server side.			✓	✓	✓
V5.6	Verify that a single input validation control is used by the application for each type of data that is accepted.				✓	✓
V5.7	Verify that all input validation failures are logged.				✓	✓
V5.8	Verify that all input data is canonicalized for all downstream decoders or interpreters prior to validation.					✓



V6.9	Verify that for each type of output encoding/escaping performed by the application, there is a single security control for that type of output for the intended destination.					✓	✓
V6.10	Verify that all code implementing or using output validation controls is not affected by any malicious code.						✓

V7 - Cryptography Verification Requirements

The Encryption Verification Requirements define a set of requirements that can be used to verify an application's encryption, key management, random number, and hashing operations. Applications should always use FIPS 140-2 validated cryptographic modules, or cryptographic modules validated against an equivalent standard (e.g., a non-U.S. standard). The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 7 - OWASP ASVS Cryptography Requirements (V7)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V7.1	Verify that all cryptographic functions used to protect secrets from the application user are implemented server side.		✓	✓	✓	✓
V7.2	Verify that all cryptographic modules fail securely.		✓	✓	✓	✓
V7.3	Verify that access to any master secret(s) is protected from unauthorized access (A master secret is an application credential stored as plaintext on disk that is used to protect access to security configuration information).			✓	✓	✓
V7.4	Verify that password hashes are salted when they are created.			✓	✓	✓
V7.5	Verify that cryptographic module failures are logged.			✓	✓	✓
V7.6	Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved random number generator when these random values are intended to be unguessable by an attacker.			✓	✓	✓
V7.7	Verify that cryptographic modules used by the application have been validated against FIPS 140-2 or an equivalent standard. (See http://csrc.nist.gov/groups/STM/cmvp/validation.html).				✓	✓



V7.8	Verify that cryptographic modules operate in their approved mode according to their published security policies (See http://csrc.nist.gov/groups/STM/cmvp/validation.html).					✓	✓
V7.9	Verify that there is an explicit policy for how cryptographic keys are managed (e.g., generated, distributed, revoked, expired). Verify that this policy is properly enforced.					✓	✓
V7.10	Verify that all code supporting or using a cryptographic module is not affected by any malicious code.						✓

V8 - Error Handling and Logging Verification Requirements

The Error Handling and Logging Verification Requirements define a set of requirements that can be used to verify the tracking of security relevant events and the identification of attack behavior. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 8 - OWASP ASVS Error Handling and Logging Requirements (V8)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V8.1	Verify that the application does not output error messages or stack traces containing sensitive data that could assist an attacker, including session id and personal information.	✓	✓	✓	✓	✓
V8.2	Verify that all server side errors are handled on the server.		✓	✓	✓	✓
V8.3	Verify that all logging controls are implemented on the server.		✓	✓	✓	✓
V8.4	Verify that error handling logic in security controls denies access by default.		✓	✓	✓	✓
V8.5	Verify security logging controls provide the ability to log both success and failure events that are identified as security-relevant.			✓	✓	✓



V8.6	Verify that each log event includes: 1. a time stamp from a reliable source, 2. severity level of the event, 3. an indication that this is a security relevant event (if mixed with other logs), 4. the identity of the user that caused the event (if there is a user associated with the event), 5. the source IP address of the request associated with the event, 6. whether the event succeeded or failed, and 7. a description of the event.				✓	✓	✓
V8.7	Verify that all events that include untrusted data will not execute as code in the intended log viewing software.				✓	✓	✓
V8.8	Verify that security logs are protected from unauthorized access and modification.				✓	✓	✓
V8.9	Verify that there is a single logging implementation that is used by the application.				✓	✓	✓
V8.10	Verify that that the application does not log application-specific sensitive data that could assist an attacker, including user's session ids and personal or sensitive information.				✓	✓	✓
V8.11	Verify that a log analysis tool is available which allows the analyst to search for log events based on combinations of search criteria across all fields in the log record format supported by this system.				✓	✓	✓
V8.12	Verify that all code implementing or using error handling and logging controls is not affected by any malicious code.						✓

V9 - Data Protection Verification Requirements

The Data Protection Verification Requirements define a set of requirements that can be used to verify the protection of sensitive data (e.g., credit card number, passport number, personally identifiable information). The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 9 - OWASP ASVS Data Protection Requirements (V9)



Verification Requirement		Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V9.1	Verify that all forms containing sensitive information have disabled client side caching, including autocomplete features.	✓	✓	✓	✓	✓	✓
V9.2	Verify that the list of sensitive data processed by this application is identified, and that there is an explicit policy for how access to this data must be controlled, and when this data must be encrypted (both at rest and in transit). Verify that this policy is properly enforced.				✓	✓	✓
V9.3	Verify that all sensitive data is sent to the server in the HTTP message body (i.e., URL parameters are never used to send sensitive data).			✓		✓	✓
V9.4	Verify that all cached or temporary copies of sensitive data sent to the client are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data (e.g., the proper no-cache and no-store Cache-Control headers are set).				✓	✓	✓
V9.5	Verify that all cached or temporary copies of sensitive data stored on the server are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.				✓	✓	✓
V9.6	Verify that there is a method to remove each type of sensitive data from the application at the end of its required retention period.					✓	✓

V10 - Communication Security Verification Requirements

The Communication Security Verification Requirements define a set of requirements that can be used to verify that all communications with an application are properly secured. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 10 - OWASP ASVS Communication Security Requirements (V10)



Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V10.1 Verify that a path can be built from a trusted CA to each Transport Layer Security (TLS) server certificate, and that each server certificate is valid.	✓		✓	✓	✓	✓
V10.2 Verify that failed TLS connections do not fall back to an insecure connection.			✓		✓	✓
V10.3 Verify that TLS is used for all connections (including both external and backend connections) that are authenticated or that involve sensitive data or functions.				✓	✓	✓
V10.4 Verify that backend TLS connection failures are logged.				✓	✓	✓
V10.5 Verify that certificate paths are built and verified for all client certificates using configured trust anchors and revocation information.				✓	✓	✓
V10.6 Verify that all connections to external systems that involve sensitive information or functions are authenticated.				✓	✓	✓
V10.7 Verify that all connections to external systems that involve sensitive information or functions use an account that has been set up to have the minimum privileges necessary for the application to function properly.				✓	✓	✓
V10.8 Verify that there is a single standard TLS implementation that is used by the application that is configured to operate in an approved mode of operation (See http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf).					✓	✓
V10.9 Verify that specific character encodings are defined for all connections (e.g., UTF-8).					✓	✓

V11 - HTTP Security Verification Requirements

The HTTP Security Verification Requirements define a set of requirements that can be used to verify security related to HTTP requests, responses, sessions, cookies, headers, and logging. The table



below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 11 - OWASP ASVS HTTP Security Requirements (V11)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V11.1 Verify that redirects do not include unvalidated data.	✓	✓	✓	✓	✓	✓
V11.2 Verify that the application accepts only a defined set of HTTP request methods, such as GET and POST.	✓	✓	✓	✓	✓	✓
V11.3 Verify that every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8).	✓	✓	✓	✓	✓	✓
V11.4 Verify that the HTTPOnly flag is used on all cookies that do not specifically require access from JavaScript.			✓	✓	✓	✓
V11.5 Verify that the secure flag is used on all cookies that contain sensitive data, including the session cookie.			✓	✓	✓	✓
V11.6 Verify that HTTP headers in both requests and responses contain only printable ASCII characters.			✓	✓	✓	✓
V11.7 Verify that the application generates a strong random token as part of all links and forms associated with transactions or accessing sensitive data, and that the application verifies the presence of this token with the proper value for the current user when processing these requests. ¹⁵					✓	✓

V12 - Security Configuration Verification Requirements

The Security Configuration Verification Requirements define a set of requirements that can be used to verify the secure storage of all configuration information that directs the security-related behavior of the application. Protection of this configuration information is critical to the secure operation of the application. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 12 - OWASP ASVS Security Configuration Requirements (V12)

¹⁵ This requirement describes the mechanism required to defend against Cross Site Request Forgery (CSRF) attacks.



Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V12.1 Verify that all security-relevant configuration information is stored in locations that are protected from unauthorized access.				✓	✓	✓
V12.2 Verify that all access to the application is denied if the application cannot access its security configuration information.				✓	✓	✓
V12.3 Verify that all changes to the security configuration settings managed by the application are logged in the security event log.					✓	✓
V12.4 Verify that the configuration store can be output in a human-readable format to facilitate audit.						✓

V13 - Malicious Code Search Verification Requirements

For Level 4, searching for malicious code in any code that has not yet been examined after performing a Level 3 application verification is required. The table below defines the Malicious Code Search requirements that are introduced at Level 4.

Table 13 - OWASP ASVS Malicious Code Search Requirements (V13)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V13.1 Verify that no malicious code is in any code that was either developed or modified in order to create the application. ¹⁶						✓
V13.2 Verify that the integrity of interpreted code, libraries, executables, and configuration files is verified using checksums or hashes.						✓

¹⁶ E.g. examine system clock calls to look for time bombs, functions unrelated to business requirements for back doors, execution paths for Easter eggs, financial transactions for incorrect logic that may indicate a salami attack, other types of malicious code.



V14 - Internal Security Verification Requirements

The Internal Security Verification Requirements define a set of requirements that can be used to verify that the application protects itself to an additional degree to guard against implementation flaws. The table below defines the corresponding verification requirements that apply for each of the four verification levels.

Table 14 - OWASP ASVS Internal Security Requirements (V14)

Verification Requirement	Level 1A	Level 1B	Level 2A	Level 2B	Level 3	Level 4
V14.1 Verify that the application protects user and data attributes and policy information used by access controls from unauthorized access or modification.					✓	✓
V14.2 Verify that security control interfaces are simple enough to use that developers are likely to use them correctly.						✓
V14.3 Verify that the application properly protects shared variables and resources from inappropriate concurrent access.						✓



Verification Reporting Requirements

An OWASP ASVS Report contains a description of the application that was analyzed against the OWASP ASVS requirements for a given level. The Report also documents the results of the analysis, including any remediation of vulnerabilities that was required.

The ASVS reporting requirements define the type of information that is required to be present in the report. The ASVS reporting requirements do not define the structure, organization, or format of the report. The ASVS reporting requirements do not preclude additional information from being included in the report.

The type of information that is required by each set of ASVS reporting requirements may be named, formatted, and organized *according to a verifier's requirements*. *The ASVS reporting requirements are met as long as the required information is present*. A Report should include all material necessary for a reader to understand the analysis that was performed and the results of the analysis, including configuration information and code snippets, as depicted in the adjacent figure, which *may* be used when constructing the Report outline.

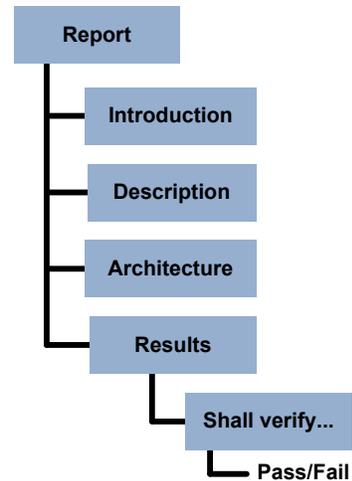


Figure 11 - Report Contents

R1 - Report Introduction

- R1.1 The report introduction shall provide sufficient information to identify both the report and the application that is the subject of the report.
- R1.2 The report introduction shall summarize the overall confidence in the security of the application.
- R1.3 The report introduction shall identify the key business risks associated with operating the application.
- R1.4 The report introduction shall identify the rules of engagement associated with performing the verification or that that may have constrained the scope of the verification.

R2 - Application Description

- R2.1 The application description shall provide sufficient description of the application to aid the understanding of its operation and the environment that it operates in.

R3 - Application Security Architecture

- R3.1 The application security architecture shall provide additional detail describing the application as the first step in providing confidence to the reader of the report that the analysis that was performed was both complete and accurate. This part of the Report provides context for the analysis. The information presented in this section will be used in the course of the analysis to identify inconsistencies. This part of the Report shall provide different levels of detail,



depending on the OWASP Application Security Verification Standard Level that the analysis was performed. Details will vary according to Level.

R4 - Verification Results

R4.1 This verification results shall present the results of the analysis that was performed according to the “Verification Requirements” section of this Standard, including description of any remediation of vulnerabilities that was required, as follows:

Table 15 - OWASP ASVS Report Verification Results Contents

Level	Pass	Fail
Level 1 Results	<ul style="list-style-type: none"> Verdict Tool configuration (if the tool can perform the check) or verdict justification (an argument for completeness and correctness, providing specific evidence) A mapping of automated tool capabilities to applicable detailed verification requirements A description of tool configuration and a mapping of tool capabilities need only be provided once as part of the report. 	<ul style="list-style-type: none"> Verdict Location (URL w/parameters and/or source file path, name and line number(s)) Description (including configuration information as appropriate) Risk rating¹⁷ Risk justification
	A description of tool configuration and a mapping of tool capabilities must also be provided as part of the report.	

¹⁷ For more information about identifying risks and estimating risks associated with vulnerabilities, see the *Testing Guide* (OWASP, 2008).



Level	Pass	Fail
Levels 2 - 4 Results	<ul style="list-style-type: none">• Verdict• Verdict justification (an argument for completeness and correctness, providing specific evidence)	<ul style="list-style-type: none">• Verdict• Location (URL w/parameters and/or source file path, name and line number(s))• Description (including path through application components and steps to reproduce)• Risk rating (see the OWASP Risk Rating Methodology)• Risk justification



Glossary

Access Control - A means of restricting access to files, referenced functions, URLs, and data based on the identity of users and/or groups to which they belong.

Application Component - An individual or group of source files, libraries, and/or executables, as defined by the verifier for a particular application.

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 ASVS.

Application Security Verification Report - A report that documents the overall results and supporting analysis produced by the verifier for a particular application.

Application Security Verification Standard (ASVS) - An OWASP standard that defines four levels of application security verification for applications.

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.

Back Doors - A type of malicious code that allows unauthorized access to an application.

Blacklist - A list of data or operations that are not permitted, for example a list of characters that are not allowed as input.

Common Criteria (CC) - A multipart standard that can be used as the basis for the verification of the design and implementation of security controls in IT products.

Communication Security - The protection of application data when it is transmitted between application components, between clients and servers, and between external systems and the application.

Design Verification - The technical assessment of the security architecture of an application.

Internal Verification - The technical assessment of specific aspects of the security architecture of an application as defined in the OWASP ASVS.

Cryptographic module - Hardware, software, and/or firmware that implements cryptographic algorithms and/or generates cryptographic keys.

Denial of Service (DOS) Attacks - The flooding of an application with more requests than it can handle.

Dynamic Verification - The use of automated tools that use vulnerability signatures to find problems during the execution of an application.

Easter Eggs - A type of malicious code that does not run until a specific user input event occurs.

External Systems - A server-side application or service that is not part of the application.

FIPS 140-2 - A standard that can be used as the basis for the verification of the design and implementation of cryptographic modules

Input Validation - The canonicalization and validation of untrusted user input.

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/>

Output Validation - The canonicalization and validation of application output to Web browsers and to external systems.

OWASP Enterprise Security API (ESAPI) - A free and open collection of all the security methods that developers need to build secure Web applications. See: <http://www.owasp.org/index.php/ESAPI>

OWASP Risk Rating Methodology - A risk rating methodology that has been customized for application security. See: http://www.owasp.org/index.php/How_to_value_the_real_risk

OWASP Testing Guide - A document designed to help organizations understand what comprises a testing program, and to help them identify the steps needed to build and operate that testing program. See: http://www.owasp.org/index.php/Category:OWASP_Testing_Project

OWASP Top Ten - A document that represents a broad consensus about what the most critical Web application security flaws are. See: <http://www.owasp.org/index.php/Top10>

Positive - See whitelist.

Salami Attack - A type of malicious code that is used to redirect small amounts of money without detection in financial transactions.

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 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).

Security Configuration - The runtime configuration of an application that affects how security controls are used.

Static Verification - The use of automated tools that use vulnerability signatures to find problems in application source code.

Target of Verification (TOV) - If you are performing an application security verification according to the OWASP ASVS requirements, the verification will be of a particular application. This application is called the "Target of Verification" or simply the TOV.

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.

Time Bomb - A type of malicious code that does not run until a preconfigured time or date elapses.

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.



Where To Go From Here

OWASP is the premier site for Web application security. The OWASP site hosts many projects, forums, blogs, presentations, tools, and papers. Additionally, OWASP hosts two major Web application security conferences per year, and has over 80 local chapters. The OWASP ASVS project page can be found here <http://www.owasp.org/index.php/ASVS>

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

- *OWASP Top Ten Project* - http://www.owasp.org/index.php/Top_10
- *OWASP Code Review Guide* - http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- *OWASP Testing Guide* - http://www.owasp.org/index.php/Testing_Guide
- *OWASP Enterprise Security API (ESAPI) Project* - <http://www.owasp.org/index.php/ESAPI>
- *OWASP Legal Project* - http://www.owasp.org/index.php/Category:OWASP_Legal_Project

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

- *OWASP* - <http://www.owasp.org>
- *MITRE* - Common Weakness Enumeration - Vulnerability Trends, <http://cwe.mitre.org/documents/vuln-trends.html>
- *PCI Security Standards Council* - publishers of the PCI standards, relevant to all organizations processing or holding credit card data, <https://www.pcisecuritystandards.org>
- *PCI Data Security Standard (DSS) v1.1* - https://www.pcisecuritystandards.org/pdfs/pci_dss_v1-1.pdf

THE BELOW ICONS REPRESENT WHAT OTHER VERSIONS ARE AVAILABLE IN PRINT FOR THIS TITLE BOOK.

ALPHA: “Alpha Quality” book content is a working draft. Content is very rough and in development until the next level of publication.

BETA: “Beta Quality” book content is the next highest level. Content is still in development until the next publishing.

RELEASE: “Release Quality” book content is the highest level of quality in a books title’s lifecycle, and is a final product.



ALPHA
PUBLISHED



BETA
PUBLISHED



RELEASE
PUBLISHED

YOU ARE FREE:



to share - to copy, distribute and transmit the work



to Remix - to adapt the work

UNDER THE FOLLOWING CONDITIONS:



Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Share Alike. - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.



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. Everyone is free to participate in OWASP and all of our materials are available under a free and open software license. The OWASP Foundation is a 501c3 not-for-profit charitable organization that ensures the ongoing availability and support for our work.

On the cover: Braconid wasps are beneficial parasites. Braconids parasitize a broad range of hosts: caterpillars, flies, wasps, beetles, and aphids. After a female injects an egg into a host, the larva feeds slowly on that single host. By the time the host dies, the larva is fully grown. It pupates inside or near the dead host, sometimes in a silken cocoon, to emerge later as an adult wasp.