

Application Security Verification Standard 3.0

May 2015

Table of Contents

[Acknowledgements 3](#_Toc419822093)

[Version 2.1, 2015 3](#_Toc419822094)

[Version 2.0, 2014 3](#_Toc419822095)

[Version 1.0 – 2009 3](#_Toc419822096)

[Copyright and License 3](#_Toc419822097)

[Introduction 3](#_Toc419822098)

[How to Use This Standard 3](#_Toc419822099)

[Application Security Verification Levels 3](#_Toc419822100)

[Level 0: Cursory 3](#_Toc419822101)

[Level 1: Opportunistic 3](#_Toc419822102)

[Level 2: Standard 3](#_Toc419822103)

[Level 3: Advanced 3](#_Toc419822104)

[Scope of Verification 3](#_Toc419822105)

[Detailed Verification Requirements 3](#_Toc419822106)

[V2: Authentication Verification Requirements 4](#_Toc419822107)

[V3: Session Management Verification Requirements 5](#_Toc419822108)

[V4: Access Control Verification Requirements 6](#_Toc419822109)

[V5: Malicious Input Handling Verification Requirements 7](#_Toc419822110)

[V7: Cryptography at Rest Verification Requirements 8](#_Toc419822111)

[V8: Error Handling and Logging Verification Requirements 9](#_Toc419822112)

[V9: Data Protection Verification Requirements 10](#_Toc419822113)

[V10: Communications Security Verification Requirements 11](#_Toc419822114)

[V11: HTTP Security Verification Requirements 12](#_Toc419822115)

[V13: Malicious Controls Verification Requirements 13](#_Toc419822116)

[V15: Business Logic Verification Requirements 14](#_Toc419822117)

[V16: Files and Resources Verification Requirements 15](#_Toc419822118)

[V17: Mobile Verification Requirements 16](#_Toc419822119)

[Appendix A: Applying ASVS in Practice 17](#_Toc419822120)

[Appendix B: Glossary 18](#_Toc419822121)

[Appendix C: Where To Go From Here 19](#_Toc419822122)

# Acknowledgements

### Version 3.0, 2015

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Leads | Andrew van der Stock | Daniel Cuthbert |  |  |
| Lead authors |  |  |  |  |
| Contributors and reviewers | Boy Baukema | Ari Kesäniemi | Colin Watson | Jim Manico |
| François-Eric Guyomarc’h | Cristinel Dumitru | James Holland | Gary Robinson |
|  |  |  |  |  |
|  |  |  |  |  |

### Version 2.0, 2014

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Leads | Daniel Cuthbert | Sahba Kazerooni |  |  |
| Lead authors | Andrew van der Stock | Krishna Raja |  |  |
| Contributors and reviewers | Antonio Fontes | Colin Watson | Jeff Sergeant | Pekka Sillanpää |
| Archangel Cuison | Dr Emin Tatli | Jerome Athias | Safuat Hamdy |
|  | Ari Kesäniemi | Etienne Stalmans | Jim Manico | Scott Luc |
|  | Boy Baukema | Evan Gaustad | Mait Peekma | Sebastien Deleersnyder |

### Version 1.0 – 2009

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Leads | Mike Boberski | Jeff Williams | Dave Wichers |  |
| Authors and contributors | Andrew van der Stock | Dr. Sarbari Gupta | John Steven | Pierre Parrend |
| Barry Boyd | Dr. Thomas Braun | Ken Huang | Richard Campbell |
|  | Bedirhan Urgun | Eoin Keary | Ketan Dilipkumar Vyas | Scott Matsumoto |
|  | Colin Watson | Gaurang Shah | Liz Fong | Shouvik Bardhan |
|  | Dan Cornell | George Lawless | Mandeep Khera | Stan Wisseman |
|  | Dave Hausladen | Jeff LoSapio | Matt Presson | Stephen de Vries |
|  | Theodore Winograd | Jeremiah Grossman | Nam Nguyen | Steve Coyle |
|  | Dave van Stein | John Martin | Paul Douthit | Terrie Diaz |

## Copyright and License

Copyright © 2008 – 2015 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.

# Preface

Our biggest goal with this version of the standard was to increase adoption.

One of the major challenges of a standard such as this is that it needs to satisfy two distinct, and very different, targets: individuals who are involved in organizing or executing a software security program within their organization, and software security professionals who conduct verification of applications. While both targets seek an industry-accepted standard for verification of applications, they operate under different constraints. For example, one of the most widely voiced criticisms of ASVS 2009 standard was that it specified automated assessments as one of the levels (or sub-levels). Many large organizations see automated assessments as a point of entry into the verification hierarchy, and thus a fully automated level is a convenient concept for them. Information security professionals, however, know that the depth and breadth of such a review will depend on what technology is used to perform the scan, thus leaving too much room for interpretation of the standard. ASVS 2014 introduces a Cursory Level 0 to allow for the flexibility needed to overcome this challenge.

On a similar note, one of the main goals for this version of the standard was to focus on the "what" and not the "how". Whereas the previous version of the standard talked about dynamic scanning, static analysis, Threat Modeling, and design reviews, you will notice that such terms do not appear in this version of the standard. Instead, we essentially define security requirements that must be verified for an application to achieve a certain level. How those requirements are verified is left up to the verifier.

Another major challenge that we overcame is to clearly separate requirements from design from scope. The previous version of the standard did not clearly distinguish between these concepts, leaving room for confusion. In this version, Level 3 is where design considerations are introduced and clearly separated from detailed verification requirements. Furthermore, we have now separated out the concept of scope completely – the new (+) notation allows for a verifier to optionally include third party components and frameworks in their review.

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.

# Introduction

The primary aim of the OWASP Application Security Verification Standard (ASVS) 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.

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 ASVS standard provides a basis for verifying 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. This standard can be used to establish a level of confidence in the security of Web applications

## How to Use This Standard

The ASVS standard can be used by both consumers and service or tool providers.

ASVS has two main goals, as depicted in the figure below: to help organization’s develop and maintain secure applications; and to allow security service/tools providers and consumers to align their requirements and offerings.

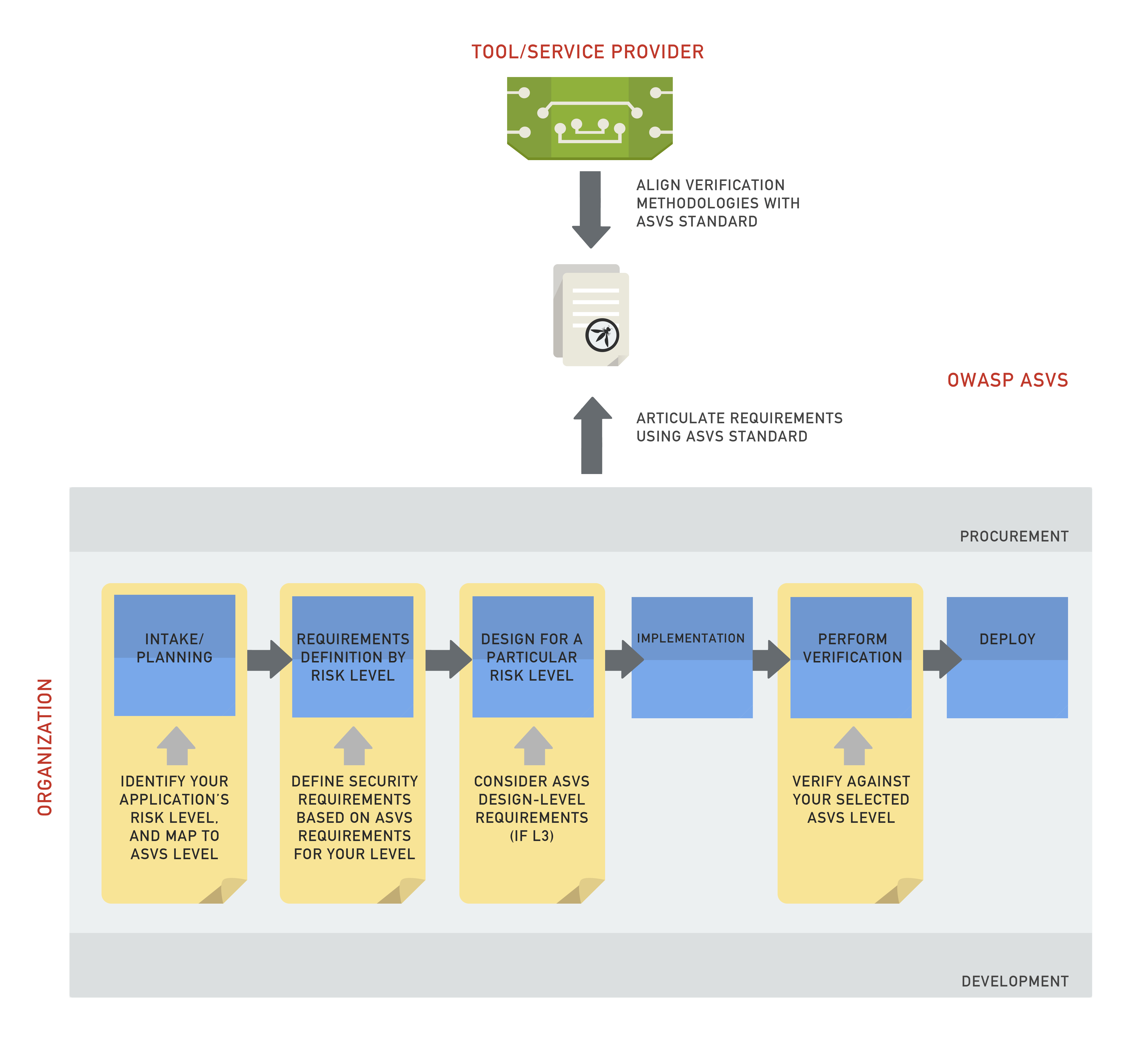


Figure - Uses of ASVS for organizations and tool/service providers

The example scenarios below further demonstrate the common use cases of ASVS using a fictional organization (ACME Bank) and a fictional security services firm (Hack All the Things).

### Use Case 1: Certification of Applications

ACME Bank has developed a new Internet Banking portal. which is due to be deployed into their production environment. The application has followed the bank's SDLC process and should be in a secure state. The internal security team at ACME Bank has been tasked to ensure that once deployed into the production environment. it does not pose a risk to other applications. due to it being hosted on a shared platform and database. After an internal threat modelling exercise was performed. it was agreed that the application had a high-risk associated with it and the data stored within it.

The team makes use of a well-known web application scanning tool and start the process of mapping out the application in preparation for the automated scanning phase. Once complete. the automated scanning tool is started and left to complete. Once the report has been generated. the security analyst tests for false positives (such as SQL injection. or X55) and amends the report as necessary. Any findings discovered are reported back to the system owners and development team. in order to be rectified. Once this has been completed. the re-test of the application is performed to ensure they have been resolved

in a suitable manner.

in this example. using the ASVS could allow the internal team to test for common application flaws as well as verify that it had been developed in accordance to the bank's security standard.

### Use Case 2 - Alignment of testing methodology

Hack All the Things (HATT) is a penetration-testing consultancy. whose main area of expertise is performing application security assessments for clients at an infrastructure and application level. They have decided to align their internal testing methodology with that of the OWASP ASVS to offer their clients peace of mind when performing assessments.

In order to achieve this. all staff ate required to manually test the application in question using the detailed verification requirements. as outlined by the ASVS document.

In this instance, adopting the ASVS allows HATT to offer a series of application assessments based on the three ASVS levels. and at the same time. allowing clients to understand what has been assessed.

### Use Case 3: Selection of external supplier

ACME Bank has finally completed all development on their new Internet Banking portal and the banking regulators require them to have an external consultancy perform an assessment of the application to ensure it meets the regulatory requirements with regards to security.

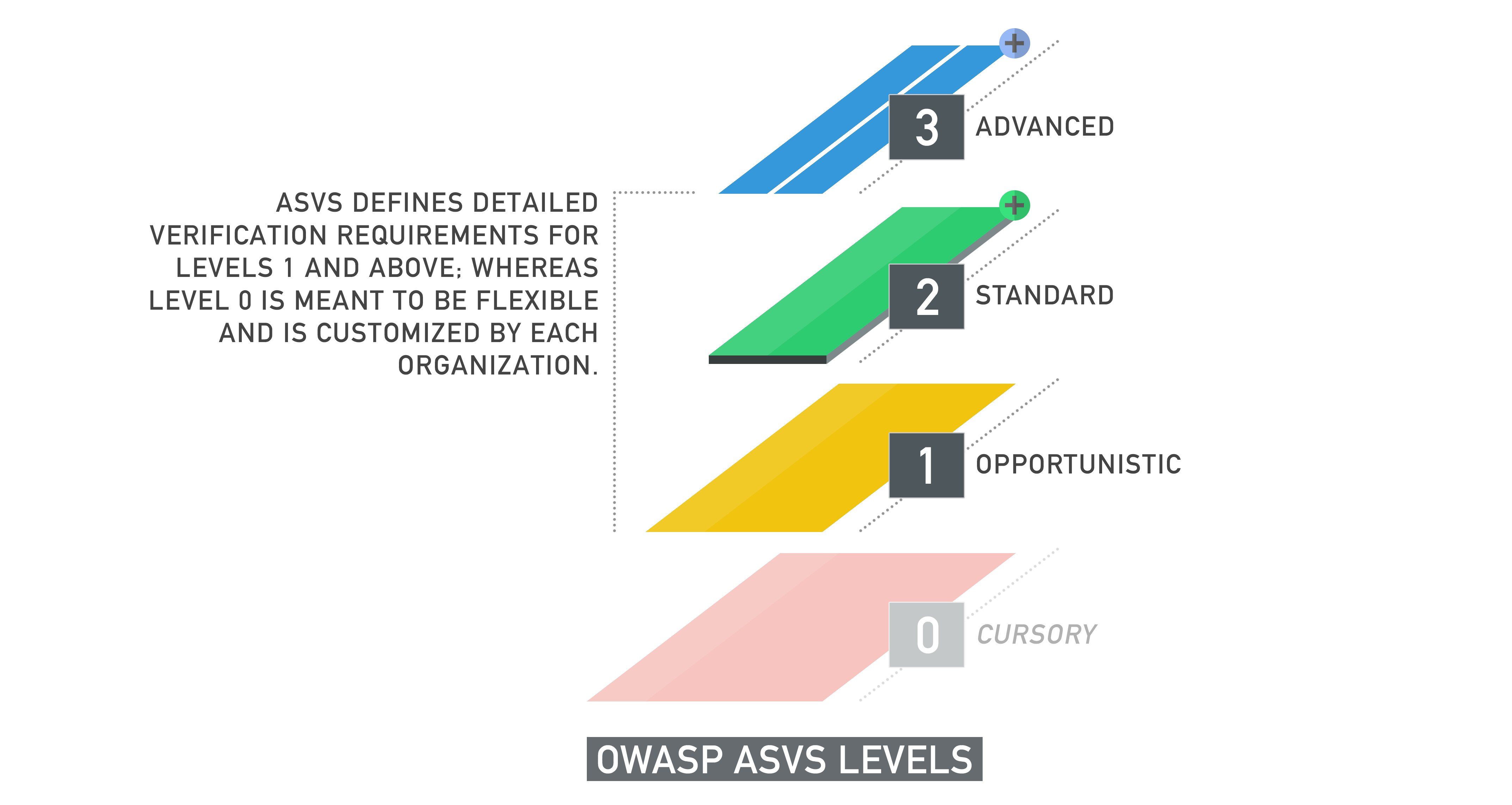
ACME Bank has chosen a supplier from their list of preferred suppliers and asked MATT to perform an assessment. ACME Bank supplied the consultancy with all the source code and documentation and scheduled the assessment. The external test was conducted in a phased approach. with a hilly-automated static analysis code review performed on the source code alongside a manual application security assessment. In addition. business logic was mated to ensure that the application performed as expected. as outlined in the functional specification documentation supplied. Once the assessment was complete. a report was created and delivered to ACME Bank staff.

By both parties adopting the ASVS during this process, the suitable level was chosen and tested for. As a result, both ACME bank and HATT were in sync with what had to be achieved and what the required outcome was.

## Application Security Verification Levels

The ASVS defines three levels of verification, with each level increasing in depth as the verification moves up the levels.

The depth is defined in each level by a set of security verification requirements that must be addressed (these are included in the requirements tables towards the end of this document). It is a verifier’s responsibility to determine if a target of verification (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.



The breadth of the verification is defined by what parts of the application are reviewed for each security requirement. For example, the scope of the review may go beyond the application’s custom-built code and include external components. Achieving a verification level under such scrutiny can be represented by annotating a “+” symbol to the verification level.

### Level 0: Cursory

Level 0 (or Cursory) is an optional certification, indicating that the application has passed some type of verification.

Level 0 is designed to be a flexible point of entry into the verification hierarchy; it indicates that some type of review has been done on the application. The detailed verification requirements are not provided by ASVS. Instead, organizations can define their own minimum criteria (such as automated runtime scan, or strong authentication mechanism).

This level is most appropriate for organizations that have a large number of applications, and where a low cost point of entry may be required. One organization may use Level 0 to require a cursory automated scan of all of their external facing applications using the organization’s commercial tool of choice; whereas another organization may define L0 requirements using data from a recent breach.

Unlike the other ASVS levels, Level 0 is not a prerequisite for other levels - an application can jump straight to Level 1 without achieving Level 0 certification (if L0 is not defined by the organization).

When defining Level 0 requirements, it is advised that each requirement be documented in a similar manner to the Detailed Verification Requirements in this document – clear, distinct, realistic, and verifiable.

Overview of Verification Requirements

### Level 1: Opportunistic

An application achieves Level 1 (or Opportunistic) certification if it adequately defends against application security vulnerabilities that are easy to discover.

The specific set of vulnerabilities against which Level 1 verification is measured is detailed in the Detailed Verification Requirements, but typically includes vulnerabilities that a verifier can identify with minimal-to-low effort. As such, this level cannot be considered a thorough inspection or verification of the application, but more of a quick inspection.

Level 1 is typically appropriate for applications where some confidence in the correct use of security controls is required, or to provide a quick sweep of a fleet of enterprise applications, to assist in developing a roadmap for more thorough inspections at a later date.

Threats to the application will most likely be from attackers who are using simple 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.

Overview of Verification Requirements

### Level 2: Standard

An application achieves Level 2 (or Standard) verification if it also adequately defends against prevalent application security vulnerabilities whose existence poses moderate-to-serious risk.

The specific set of vulnerabilities against which Level 2 verification is measured is detailed in the Detailed Verification Requirements, but would include OWASP Top 10 vulnerabilities and business logic vulnerabilities.

Level 2 ensures that evaluated security controls are in place, effective, and used as needed within the application to enforce application-specific policies.

Level 2 represents an industry standard for which the majority of an organization’s sensitive applications would strive. 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 security will typically be opportunists and possibly determined attackers (skilled and motivated attackers focusing on specific targets using purpose-built scanning tools as well as manual testing techniques).

Overview of Level 2 Verification Requirements

Level 2 Applications are assessed according to the following criteria:

L2.1 Controls are assessed and determined to be in place, in use and effective

L2.2 Controls are assessed such that false negatives and false positives are removed from any automated results

### Level 3: Advanced

An application achieves Level 3 (or Advanced) certification if it also adequately defends against all advanced application security vulnerabilities, and also demonstrates principles of good security design. Level 3 is at the very least, is a source code review. The best method of reviewing an application at Level 3 is with both the code and the source code.

The specific set of vulnerabilities against which Level 3 verification is measured is detailed in the Detailed Verification Requirements, but would include more difficult to exploit vulnerabilities, which would most likely be exploited by determined attackers.

Level 3 is the only ASVS level which also requires an inspection of the application’s design. In addition, the following requirements were added:

- Any major security controls which have a cross-cutting impact (such as input validation or authorization) should be implemented in a centralized manner.

- Security controls that perform validation should make decisions using a whitelist (“positive”) approach.

- Input validation should not be relied on as the only defense against injection and scripting vulnerabilities. Rather, input validation should always be the second line of defense, with parameterization and output encoding being the primaries, respectively.

Level 3 verification is typically appropriate for critical applications that protect life and safety, critical infrastructure, or defense functions or have the potential of facilitating substantial damage to the organization. Level 3 may also be appropriate for applications that process sensitive assets.

Threats to security will be from determined attackers (skilled and motivated attackers focusing on specific targets using tools including purpose-built scanning tools).

Overview of Level 3 Verification Requirements

L3.1 Application is assessed according to the Level 3 requirements in each of the applicable detailed verification sections

L3.2 Application is verified that implementation of all security controls adhere to the following leading practices:

• Security controls that perform validation make decisions using a positive (“whitelist”) approach

• Data validation controls are complemented by contextually aware output encoding

• All untrusted data that is sent to database layer interpreters either use parameterized interfaces, object relational models not subject to SQL injection, or are otherwise rendered safe for the data access layer.

L3.3 Application reviews are primarily manual with automated assistance for coverage and identification purposes. Controls should be assessed to the following increased minima:

• Consider if identified security controls are designed to be fit for purpose or could be improved

• Security controls are in place, in use and effective

• Security controls are centralized within the application

• All automated results are reviewed in detail, ensuring that coverage is sufficient well above sampling approaches, and that false positive and negatives results are investigated via inspection and/or discussion with the project team

## Scope of Verification

The scope of the verification is separate from the requirements for achieving a level.

Be default, ASVS assumes that the scope of the verification includes all code that was developed or modified in order to create the application or release. However, one may decide to include as part of verification the code for all third-party frameworks, libraries, and service security functionality that is invoked by or supports the security of the application. Achieving a verification level under such scrutiny can be represented by annotating a “+” symbol to the verification level. For example, an application may be labelled as ASVS L3+ certified.

Including third party components is optional and is not required to achieve to any ASVS level. Such level of scrutiny may be suitable for highly sensitive or mission critical applications. As such, (+) certification will in most cases be associated with Level 3.

When third party components are included in the verification, it is not required that all detailed verification requirements be applied to third party components. In fact, most detailed verification requirements will not be applicable to third party components and can thus be checked against the base code only. Detailed verification requirements must be verified against the application’s base code, and they are verified against third party components if applicable. Only then can an application achieve the (+) certification for that level.

# Detailed Verification Requirements

V2. Authentication

V3. Session Management

V4. Access Control

V5. Malicious Input Handling

V6. Client Side Security (NEW for 3.0)

V7. Cryptography at Rest

V8. Error Handling and Logging

V9. Data Protection

V10. Communications

V11. Configuration

V13. Malicious Controls

V15. Business Logic

V16. File and Resource

V17. Mobile

V18. Web services (NEW for 3.0)

V19. Restful services (NEW for 3.0)

# V2: Authentication Verification Requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V3: Session Management Verification Requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V4: Access Control Verification Requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V5: Malicious input handling verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V7: Cryptography at rest verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V8: Error handling and logging verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V9: Data protection verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V10: Communications security verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V11: Security configuration verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V13: Malicious controls verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V15: Business logic verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

Verify the application has business limits and enforces them in a trusted location (as on a protected server) on a per user, per day or daily basis, with configurable alerting and automated reactions to automated or unusual attack. Examples include (but not limited to): ensuring new SIM users don’t exceed $10 per day for a new phone account, a forum allowing more than 100 new users per day or preventing posts or private messages until the account has been verified, a health system should not allow a single doctor to access more patient records than they can reasonably treat in a day, or a small business finance system allowing more than 20 invoice payments or $1000 per day across all users. In all cases, the business limits and totals should be reasonable for the business concerned. The only unreasonable outcome is if there are no business limits, alerting or enforcement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V16: Files and resources verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V17: Mobile verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V18: Web services verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V19. Configuration

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# V20. Client side verification requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Description | 1 | 2 | 3 | Since |
| V20.20 |  |  |  |  | 1.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Appendix A: Applying ASVS in Practice

Different threats have different motivations, and some industries have unique information and technology assets as well as 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 vulnerable applications reachable through the Internet, which is why ASVS Level 1 is recommended for all Internet-accessible applications regardless of industry. This is a suggested starting point, considering a small number of risk factors. 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 SEGMENT | THREAT PROFILE | SUGGESTED ASVS LEVEL |
| --- | --- | --- |
| 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. | Level 1: all Internet-accessible applications. |
|  | Some major compliance considerations include Payment Card Industry Data Security Standard (PCI DSS), Gramm-Leech Bliley act, Sarbanes Oxley (SOX). | Level 2: applications that contain sensitive information like credit card numbers, personal information, can move limited amounts of money in limited ways. Examples include: (i) transfer money between accounts at the same institution or (ii) a slower form of money movement (e.g. ACH) with transaction limits or (iii) wire transfers with hard transfer limits within a period of time. |
|  |  | Level 3: applications that contain large amounts of sensitive information or that allow either rapid transfer of large sums of money (e.g. wire transfers) or transfer of large sums of money in the form of individual transactions or as a batch of smaller transfers. |
| Manufacturing, Professional, Transportation, Technology, Utilities, Infrastructure, Defense | These industries may not appear to have very much in common, but the threat actors who are likely to attack organizations in this segment are more likely to perform focused attacks with more time, skill, and resources. Often the sensitive information or systems are not easy to locate and require leveraging insiders and social engineering techniques. Attacks may involve insiders, outsiders, or be collusion between the two. Their goals may include gaining access to intellectual property for strategic or technological advantage. We also do not want to overlook attackers looking to abuse application functionality influence the behaviour of or disrupt sensitive systems. | Level 1: all Internet-accessible applications. |
|  | Most attackers are looking for sensitive data that can be used to directly or indirectly profit from to include personally identifiable information and payment data. Often the data can be used for identity theft, fraudulent payments, or a variety of fraud schemes. | Level 2: applications containing internal information or information about employees that may be leveraged in social engineering. Applications containing non-essential, but important intellectual property or trade secrets. |
|  |  | Level 3: applications containing valuable intellectual property, trade secrets, or government secrets (e.g. in the United States this may be anything classified at Secret or above) that is critical to the survival or success of the organization. Applications controlling sensitive functionality (e.g. transit, manufacturing equipment, control systems) or that have the possibility of threatening safety of life. |
| Healthcare | Most attackers are looking for sensitive data that can be used to directly or indirectly profit from to include personally identifiable information and payment data. Often the data can be used for identity theft, fraudulent payments, or a variety of fraud schemes. | Level 1: all Internet-accessible applications. |
|  |  | Level 2: applications with small or moderate amounts of sensitive medical information (Protected Health Information), Personally Identifiable Information, or payment data. |
|  |  | Level 3: Applications used to control medical equipment, devices, or records that may endanger human life. Payment and Point of Sale systems (POS) that contain large amounts of transaction data that could be used to commit fraud. This includes any administrative interfaces for these applications. |
| Retail, Food, Hospitality | Many of the attackers in this segment utilize opportunistic "smash and grab" tactics. However, there is also a regular threat of specific attacks on applications known to contain payment information, perform financial transactions, or store personally identifiable information. Although less likely than the threats mentioned above, there is also the possibility of more advanced threats attacking this industry segment to steal intellectual property, gain competitive intelligence, or gain an advantage with the target organization or a business partner in negotiations. | Level 1: all Internet-accessible applications. |
|  |  | Level 2: Suitable for business applications, product catalogue information, internal corporate information, and applications with limited user information (e.g. contact information). Applications with small or moderate amounts of payment data or checkout functionality. |
|  |  | Level 3: Payment and Point of Sale systems (POS) that contain large amounts of transaction data that could be used to commit fraud. This includes any administrative interfaces for these applications. Applications with a large volume of sensitive information like full credit card numbers, mother's maiden name, social security numbers etc. |

# Appendix B: 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.
* **Certificate Authority** (CA) – An entity that issues digital certificates.
* **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.
* **Cross-Site Scripting** (XSS) – A security vulnerability typically found in web applications allowing the injection of client-side scripts into content.
* **Cascading Style Sheets** (CSS) - A style sheet language used for describing the presentation semantics of document written in a markup language, such as HTML.
* **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
* **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.
* **HTML** – The main markup language for the creation of web pages and other information displayed in a web browser.
* **Input Validation** – The canonicalization and validation of untrusted user input.
* **LDAP** – An application protocol for accessing and maintaining distributed directory information services over a network.
* **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 encoding** – 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** **validation** – 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.
* **SQL Injection (SQLi)** – A code injection technique used to attack data driven applications, in which malicious SQL statements are inserted into an entry point.
* **Target of Verification (TOV)** – If you are performing 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.
* **Transport Layer Security** – Cryptographic protocols that provide communication security over the Internet
* **User acceptance testing (UAT)**– Traditionally a test environment that behaves like the production environment where all software testing is performed before going live.
* **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.
* **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.
* **XML** – A markup language that defines a set of rules for encoding documents.

# Appendix C: Where to go from here

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.

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 Code Review Guide - <http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project>
* 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>
* OWASP Testing Guide - <http://www.owasp.org/index.php/Testing_Guide>
* OWASP Top Ten Project - <http://www.owasp.org/index.php/Top_10>

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 - <http://cwe.mitre.org/>
* 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) v3.0 Requirements and Security Assessment Procedures <https://www.pcisecuritystandards.org/documents/PCI_DSS_v3.pdf>