OWASP
Open Web Application Security Project

Application Security Verification Standard 4.0

Andrew van der Stock, co-leader

May 2019

Andrew van der Stock

- Senior Principal Consultant, Synopsys
  - Technical Leader of Managed Services
- Joined OWASP late ~2002
  - Lifetime OWASP member
  - Board Member (2015-2018) and Treasurer (2016-2018)
- Selected works:
  - Application Security Verification Standard
  - OWASP Top 10 {2007, 2017}
  - OWASP Developer Guide 2.0

What is the ASVS?

- Started as 80/20 checklist
- Designed to be an actual application security standard
- Set of leading practices – even 2.0 was challenging for many
- Community and Industry Driven

- Completely developed in the open at GitHub
  - Submit issues! Submit PRs! Translate please!

Credential service providers (CSPs) provide federated identity for users. Users will often have more than one identity with multiple CSPs, such as an enterprise identity using Azure AD, Okta, Ping Identity or Google, or consumer identity using Facebook, Twitter, Google, or WeChat, to name a just few common alternatives. This list is not an endorsement of these companies or services, but simply an encouragement for developers to consider the reality that many users have many established identities. Organizations should consider integrating with existing user identities, as per the risk profile of the CSP's strength of identity proofing. For example, it is unlikely a government organization would accept a social media identity as a login for sensitive systems, as it is easy to create fake or throw away identities, whereas a mobile game company may well need to integrate with major social media platforms to grow their active player base.

#	Description	L1	L2	L3	CWE	NIST §
2.1.1	Verify that user set passwords are at least 12 characters in length. (C6)	✓	✓	✓	521	5.1.1.2
2.1.2	Verify that passwords 64 characters or longer are permitted. (C6)	✓	✓	✓	521	5.1.1.2
2.1.3	Verify that passwords can contain spaces and truncation is not performed. Consecutive multiple spaces MAY optionally be coalesced. (C6)	✓	✓	✓	521	5.1.1.2
2.1.4	Verify that Unicode characters are permitted in passwords. A single Unicode code point is considered a character, so 12 emoji or 64 kanji characters should be valid and permitted.	✓	✓	✓	521	5.1.1.2
2.1.5	Verify users can change their password.	✓	✓	✓	620	5.1.1.2
2.1.6	Verify that password change functionality requires the user's current	✓	✓	✓	620	5.1.1.2

Who is involved?

- You

- Andrew van der Stock, Daniel Cuthbert, Jim Manico
- Josh Grossman, Mark Burnett, Abhay Bhargarv
- Amazing reviewers such as Elar Lang, ossie-git, Ron Perris, Tonimir Kisasondi, Serg Belokamen, Jason Axley, and Adam Caudill

SO WHAT'S NEW?

What's new

- Now completely written in Markdown
  - Uses MASVS script and CSV generation
  - Easy to translate
  - Easy to determine what changed, when, by whom, and why
- NIST 800-63 compliance
- Data Protection has been upped to be primarily about human sensitive personal identifying information
  - Helps with GDPR and APPs
- IoT ASVS Preview Chapter

Modern web applications

- Full support for server-less, responsive applications
- Containers
- API
- DOM
- Templating

CWE all the things

- Most requested feature for the last decade finally delivered
- Let's talk about CWE for a minute
  - ASVS is a control based standard
  - Weaknesses are not controls
  - CWE is an imperfect mapping, but it's the mapping we have
- Not every item ended up with a CWE. CWE needs our help

What's changed

- Basically everything

- Renumbered completely
- Each section is reorganized and re-ordered
- De-duped. Do not omit any section for your level

L1 is the new minimum

- OWASP Top 10 2017 is simply not sufficient

- Level 1 is now completely testable using pentest techniques
- It’s the only level that is completely penetration testable

- Stop penetration testing. Hybrid reviews at least!

PCI DSS 6.5.x

- Yep

- We even included buffer overflows, integer and safer string operations, as well as ensuring that folks compiled code properly!

What's gone

- Less impactful controls
- Controls that were implemented by one browser or language
- "Since"

- Mobile Chapter (use MASVS)
- IoT Chapter (use IoT Project)

DETAILED CHANGES

Architecture

- Completely new

- Replaces “dead” section with something that can produce secure by default software
- Design and build security in!

- Level 2 and 3 only

Authentication and NIST 800-63

- Aligned with NIST 800-63
  - Except that we had to go to 12 characters for SFA passwords
- Credential lifecycle from issuance to retirement
- Credentials construction and protection, including credential stuffing
- Multi-factor is now expected, crypto devices, web services
- JWT, Oauth, federated security
- Advanced session management attacks
  - Half open attack

Validation, Sanitization, and Encoding

- Major revamp
- Divided into easy to consume sections
  - Standardized Input and Output Pipelines
  - Input validation for modern applications and APIs
  - Output encoding is the new hotness
  - Injection prevention (including XXE and XML attacks)
  - SSRF
  - Deserialization

Communications Security

- Now DevSecOps friendly
  - No more building a secure chain / path … what does that even mean?

- Easier to comply … automatically
- TLS all the things
- Use modern configuration builders
- Use the verification tools you use today

Malicious Code

- Major update to this section to cover more than just time bombs and Easter eggs
  - Detect malicious code introduction
  - Continuous detection through building
  - Privacy invading libraries (Google's PUA) mentioned for first time
  - Malicious business logic, such as salami attacks
  - Privacy invading permissions (camera, location, microphone, contacts, etc)

- Mostly Level 3 for apps that can kill you or run the world economy

Business Logic Verification

- Major update
  - Business logic step order
  - Business logic human time
  - Business logic limits
  - Business logic anti-automation
  - Threat model (attack driven design) business logic risks
  - TOCTOU Race conditions that might affect business logic
  - Monitoring, alerting, and detection of unusual business logic events

Files and Resources

- Major revamp
  - Architecture items moved to architecture
  - Configuration items moved to configuration
- Upload - Size and number
- Contents and integrity
- Storage, including malicious file detection
- Execution, including LFI, RFI, and SSRF
- Download

API Security

- Total revamp

- General controls include common sense API controls
  - Must be read in conjunction with authentication, authorization and session management
- RESTful includes schema validation, CORS and origin attacks
- SOAP – fewer more impactful items, far clearer, and not obfuscated
- GraphQL and Web Service Data Layer

Configuration

- Major revamp
- Repeatable, Continuous integration, continuous deployment
- Support for DevSecOps culture and agile practices
- Containers
- Dependency checks mandated
- Sandboxing components including uploaded assets
- Sub-domain takeover

I'M IN. HOW DO I USE IT?

When can I get it?

- The ASVS 4.0 Final is available now!

- OWASP Wiki – Word, PDFs, CSVs, and Hot Linkable markdown
- GitHub - Final Version is in the 4.0 branch
- GitHub – Development Version is in the master branch

- You can also get this presentation so you can give this to your local chapter, school, college, or workplace!

Generally Accepted Security Practices

Secure code review

Security architecture

Integration testing

Peer coding checklists

Developer training

Unit testing

Hybrid reviews

DevSecOps automation

Vulnerability programs

Consultant training

Tool Benchmark

Secure coding checklist

Penetration test

Deployment checklist

Functional constraints

Planning Sprint Assistance

Non-functional and functional features

Supplier Benchmarking

How do I use it?

- Use it as is or fork it

- Level 1 – entry level, penetration testing
- Level 2 – most apps
- Level 3 – apps that can kill you or run the world economy

- Deep standards have no bounds

How to get involved

- Grab a copy today and start to migrate from earlier versions and T10

- We need translations. Please join the #asvs channel on OWASP Slack
- We need case studies! If you use ASVS, we'd love to hear from you!

- Create issues or pull requests if you identify issues

vanderaj@owasp.org (ASVS) | vander@synopsys.com ($dayjob)

@vanderaj

THANK YOU