The following were all the comments that we specifically tracked and made updates to the Top 10 based on these comments. There may be a few minor comments that were provided that caused us to make small updates to the Top 10 Final release but we don’t believe we have missed any significant contributions to the Top 10 for 2013.

1. The community suggested that we make the vulnerability statistics public.

We requested, and received public links from all the data suppliers. Links to each providers data set is now included in the Final Top 10 in the Acknowledgements section.

1. Christian Heinrich noted that 2 of the vulnerability statistics links (WhiteHat and HP) required user registration to access the stats.

We were able to get direct, completely public links to both reports and so now user registration is not required for these as well.

1. The community suggested we acknowledge past contributors

We added an acknowledgement that the Top 10 wouldn’t be what it is today without all the contributions to previous versions. Key contributors to each version of the Top 10, including this version, is provided in each version of the Top 10 that was released.

1. Neil Smithline posted this comment:

I think that A8 Tech Input needs some cleanup. The current text is:

Attackers can cause victims to change any data the victim is allowed to change or perform any other function the victim is authorized to use, including state changing requests, like logout or even login.

I whipped up the revised paragraph below.

Attackers can trick victims into performing any operation the victim is authorized to perform. This can include changing account email addresses, making purchases, or user login and logout.

We updated that section to say: Attackers can trick victims into performing any state changing operation the victim is authorized to perform, e.g., updating account details, making purchases, logout and even login.

1. Neil Smithline provided this comment: I think that the paragraph at the start of "What's Next ..." looks ugly because of its length. I think you can split the paragraph either before or after the sentence that begins "OWASP recommends ..."

Done

1. Neil Smithline also provided this comment: “A9 only refers to components. But I don't see why that is the case. A9 uses Apache as an example. Apache is a platform, albeit a smaller one than Linux or Windows, but still a platform. I don't see a clear delineation of "components" and "OS".”

The use of this Apache example is potentially confusing since most people think of the Apache app server when the see Apache. However, Apache CXF is a framework/library, not the Apache App Server itself. To help clarify, we rewrote this example to say: “Apache CXF Authentication Bypass – By failing to provide an identity token, attackers could invoke any web service with full permission. (Apache CXF is a services framework, not to be confused with the Apache Application Server.)”

1. Neil Smithline provided this comment:

I know there's been discussion about having mass assignment as one of the T10. The decisions was no. I'm not trying to reopen that discussion and don't feel that I have enough data to even form an opinion.

The above notwithstanding, I think it would be better if mass assignment were mentioned someone in the T10. Delegated to being just an extra risk at the end of the document gives it a much smaller presence as one must assume that additional risks gets less attention than the T10.

I'm wondering if mass assignment could be mentioned in A9. I think it could fit in the example section or in the references section with a mass reference link.

* Response: This is a good idea. We added the following reference to A9: [Example Mass Assignment Vulnerability that was fixed in ActiveRecord, a Ruby on Rails GEM](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-0277), which is to an article that you suggested to us in a separate email during our discussion of this comment.

1. Jeremiah Grossman provided the following comments:

“1) Found the following two sentences, one says "eleventh" and the other "tenth," this should be reconciled.

"This release of the OWASP Top 10 marks this project’s eleventh year of raising awareness of the importance of application security risks."

"This release of the OWASP Top 10 marks this project’s tenth year of raising awareness of the importance of application security risks."

2) This sentence implies that WhiteHat falls into the "tool" vendor category, but we're really not, nor are we consultants. We're SaaS, and I'd suspect Veracode might have the same issue as well.

"The OWASP Top 10 is based on risk data from 8 firms that specialize in application security, including 4 consulting companies and 4 tool vendors (2 static and 2 dynamic)."

I'd recommend nixing the last part entirely, as it wouldn't seem to matter what types of appsec vendors we all are in this context anyway. No need for confusion For example...

"The OWASP Top 10 is based on risk data from 8 firms that specialize in application security."

3) I know I risk holy war with this comment, but IMHO, the following statements are subjective. And to my mind unnecessarily pit the two different software security testing methodologies against each other.

"Reviewing the code is the strongest way to verify whether an application is secure. Testing can only prove that an application is insecure."

This is especially confusing because earlier in the document we find this more reasonable suggestion:

"OWASP recommends a combination of security code review and application penetration testing whenever possible, as that allows you to leverage the strengths of both techniques, and the two approaches complement each other."

Perhaps instead some comments are in order about what these testing methodology are best suited to measure / find, what circumstances they are best used, and when. Often one can and should be used over another in certain circumstances, while bother is others. Additional thoughts:

Black Box vs White Box. You are doing it wrong.

http://jeremiahgrossman.blogspot.com/2009/10/black-box-vs-white-box-you-are-doing-it.html

If this were only the case, as compliance does not really allow organizations to move beyond it.

"We encourage you to use the Top 10 to get your organization started with application security."

4) The entire document emphasizes and encourages a process driven approach to appsec throughout the pages, particularly with ASVS being recommended all over. Then there's this statement, which seems to contradict:

"These programs come in all shapes and sizes, and you should avoid attempting to do everything in a process model. Instead, leverage your existing organization’s strengths and measure what works for you."

Am I misunderstanding something. Is ASVS not a process?

5) • Broken Authentication and Session Management moved up in prevalence based on our data set,. Probably because this area is being looked at harder, not because issues are actually more prevalent. This caused Risks A2 and A3 to switch places."

I'd like to get additional insight into what "data set" and thought process when into this decision. Not saying that it's "wrong" yet, just wanna know why someone thinks so.”

Response: Suggestion 1 was fixed as requested. For 2) we reworded it to say 7 vendors and acknowledge that the tool vendors were either tool or SaaS vendors. For 3) We rewrote the intro to Code Review to acknowledge that both code review and pen testing have strengths and weaknesses and that a combined approach is frequently best. For 4) He asks is ASVS not a process? And the answer is no, ASVS is not a process. It’s a way of performing and reporting verification results. 5) Based on the metrics we were provided by the 7 providers, Authentication and Session management vulnerability prevalence seemed to go up as compared to the 2010 stats.

We provided the updated version of the Top 10 to Jeremiah to review, and he agreed the updates addressed his comments.

1. Dirk Wetter suggested we ask that people consider Content Security Policy (CSP) as a defense against XSS.

This was added to A3 (XSS).

1. Mike Boberski made the following comments:

Just doing a quick initial scan,

Re “Using Components with Known Vulnerabilities”,

Re “How Do I Prevent This”,

Perhaps consider mentioning FIPS 140-2 in order to use known good crypto, add reference links for CMVP module validation lists and SP 800-131A.

Perhaps adding reference but not going into any detail about CC VPL.

Perhaps consider reversing the order of your 3 step recommendations, updating what is currently #1 to “Identify all the components…” I.e., refine this to first get policy and procedures around major components, and chip away at minor components that aren’t packaged as part of a larger component to the extent possible.

Perhaps consider simply replacing what is currently step 1 of your recommendations with suggestion to develop secure coding guidance or reusable wrappers for using components w/issues when issues are discovered, vs. not doing things like branching from the baseline for some underlying component and rolling one’s own fixes.

Mike

Response: We think the FIPS 140 comment actually applies better to A6, the Crypto requirement.

Prevention item #3 in A6 has been updated to say:

1. Ensure strong standard algorithms and strong keys are used, and proper key management is in place. Consider using [FIPS 140 validated cryptographic modules](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm).

Regarding reordering the 3 steps in A9. we agree that would be ideal, but people aren’t likely to do #3 but are more likely to do #1 and #2 as first steps, which is why they are listed in that order.

We added a 4th idea related to your comment about use of wrappers:

1. Where appropriate, consider adding security wrappers around components to disable unused functionality and/ or secure weak or vulnerable aspects of the component.
2. Hasan Mirjalili made the following comment:

I am confused about these two sentences:

Page 2: This release of the OWASP Top 10 marks this project’s tenth year of raising awareness of the importance of application security risks.

page 3: This release of the OWASP Top 10 marks this project’s eleventh year of raising awareness of the importance of application security risks.

Is that an error?

Response: On page 3, we changed the wording to: … marks this project’s tenth anniversary of raising awareness …

Page 2 is deleted from the final release as it’s a note about the Release Candidate only.

1. Torsten Gigler provided the following comment:

Hi Dave,

I have another suggestion/discussion for A1 Injection ;-)

• How Do I prevent 'Injection': Should we give here some additional advise what to do if special characters are needed (see green background)? [If yes: Perhaps this advise could be used in other risks like XSS, too (where (manual) whitelisting is mentioned)]

Kind regards, Torsten

Response: Both suggestions 1 and 2 above #3 provide defenses if special characters are needed. We updated para 3 to say:

3. Positive or “white list” input validation is also recommended, but is not a complete defense as many applications require special characters in their input. If special characters are required, only approaches 1. and 2. above will make their use safe. OWASP’s ESAPI has an extensible library of white list input validation routines.

1. Torsten Gigler also provided this comment:

Several ones start with some sentences that would fit into 'How do I Prevent This'? The type of logic changes between positive and negative wordings.

For example A2 Authentication and Session Management:

Under the Headline 'Am I Vulnerable to..' it starts with the primary assets to protect.... May I suggest to change the wording that it belongs better to the headline:

Let me try: "Lack of protection of the primary assets credentials and session IDs:"

1. credentials are not always protected when stored not (effectively) using hashing or encryption, see A6

...

A3 XSS: If you do not ensure that all user supplied input...

A5: If you have forgotten the proper hardening across the entire application stack (one of the following is missing):

A6: If you have not determined which data is sensitive ... If you are not sure that you did the following to protect such kind of data.

A8: To check whether an application is vulnerable, see if a link or a form lacks of an unpredictable token.

Response:

For A2: We changed the intro sentence to be:

Are session management assets like user credentials and session IDs properly protected?

For A3: The beginning of the 1st paragraph was modified to say:

You are vulnerable if you do not ensure that all user supplied input sent back to the browser is properly escaped, or you do not verify it to be safe via input validation before it is included in the output page. Proper output encoding ensures that such input is always treated as text in the browser, rather than active content. If AJAX is being used to dynamically update the page, are you using safe JavaScript APIs?

For A5: This section has now been rewritten as follows:

Is your application missing the proper security hardening across any part of the application stack? Including:

1. Is any of your software out of date? This includes the OS, Web/App Server, DBMS, applications, and **all code libraries (see new A9)**.
2. Are any unnecessary features enabled or installed (e.g. ports, services, pages, accounts, privileges)?
3. Are default accounts and their passwords still enabled and unchanged?
4. Does your error handling reveal stack traces or other overly informative error messages to users?
5. Are the security settings in your development frameworks (e.g., Struts, Spring, ASP.NET) and libraries not set to secure values?

A6: This section has now been rewritten as follows:

For all such data:

1. Is any of this data stored in clear text long term, including backups of this data?
2. Is any of this data transmitted in clear text, internally or externally? Internet traffic is especially dangerous.
3. Are any old / weak cryptographic algorithms used?
4. Are weak crypto keys generated, or is proper key management or rotation missing?
5. Are any browser security directives or headers missing when sensitive data is provided by / sent to the browser?

A8: We changed the first sentence to: To check whether an application is vulnerable, see if any links and forms lack an unpredictable token.

1. Torsten and the project exchanged several additional emails and more minor updates were made based on his additional input and him reviewing the updates we made based on his original comments.
2. Adam Baso provided details comments in a serious of 3 emails. The comments and the response to each is below:

Here's batch 1. Will try to send more when I wake up tomorrow.

Double check every hyperlink. Additionally, convert all redirects to direct links, and use HTTPS where it doesn't cause browser warnings

A: If you have any specific recommendations, please let me know. We don’t have time to check them all at this level of detail.

COVER PAGE:

\* Change the URL to be HTTPS.  A: Done.

ABOUT OWASP:

\* In the Forward column, change "their enterprise" to "their enterprises".  A: We think this is one of those 6 of this vs. half a dozen of that. Each Executive probably only works for one enterprise so we think leaving it unchanged is probably best.

\* In the Forward column, change "These programs come in all shapes and sizes, and you should avoid attempting to do everything in a  process model. Instead, leverage your existing organization’s

strengths and measure what works for you." to "These programs come in all shapes and sizes. Leverage your existing organization's strengths and measure what works for you." A: We think it’s important to caution people not to try to do everything including the kitchen sink. We modified the last part of the sentence to say: “Instead, leverage your existing organization’s strengths to do and measure what works for you.”

\* In the Forward column, change "Please don’t hesitate to contact OWASP with your questions, comments, and ideas, either publicly to [owasp-topten@lists.owasp.org](mailto:owasp-topten@lists.owasp.org) or privately to

[dave.wichers@owasp.org](mailto:dave.wichers@owasp.org)" to "Please share your thoughts on the OWASP Top 10 at our public mailing list at <https://lists.owasp.org/mailman/listinfo/owasp-topten> and the OWASP Top 10 project website at <https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project>."  A: Some people prefer not to provide public comments, so we think its fine the way it is.

\* In the About OWASP column, change "security code review" to the more conventional "secure code review". A: Done

\* In the About OWASP column, replace the last two bullets with a non-bulleted sentence: "Learn more at <https://www.owasp.org>." A: Done

\* In the About OWASP column, change "open-source" to "open source". A: Done

INTRODUCTION

\* In the Welcome section, change "broadens of of categories" to "broadens one of the categories". A: Done

\* In the Warnings section, change "how to effectively find vulnerabilities in web applications are" to "how to effectively find vulnerabilities in web applications is". A: Done

\* In the Welcome section, remove the text ", which have both been significantly updated since the previous release of the OWASP Top 10". I may be mistaken here, but it seems updates may have not been instituted fully in one or both documents. Please forgive me if my search engine-fu is off tonight. A: Agree, Done. We also added a reference to the OWASP cheat sheet series.

\* In the Welcome section, change "as new flaws are discovered" to "as new flaws are discovered and attack methods are refined." A: Done

\* In the Acknowledgements section, change "Thanks to Aspect Security for initiating, leading, and update the OWASP Top 10 since its inception in 2003, and to its primary authors: Jeff Williams and Dave Wichers." to "Thanks to Jeff Williams and Dave Wichers at Aspect Security for project initiation, leadership, and continued updates for the OWASP Top 10 since its inception in 2003." A: More than just Jeff and I contribute to the Top 10 from Aspect so we would prefer to leave this as is.

\* In the Acknowledgments section, remove the Aspect Security logo.  A: We would prefer to leave this as is until, and if, the OWASP Board officially changes the policy on this.

\* Change "We would also like to thank everyone who contributed to previous versions of the Top 10, without which, it wouldn't be what it is today." to "We would also like to thank everyone who contributed to previous versions of the Top 10. Without these contributions, the Top 10 wouldn't be what it is today." A: Done

\* Add names and organizations for individuals who contributed. A: Done

RELEASE NOTES

\* In the What Changed From 2010 to 2013? section, change "date set,." to "data set.". A: Done

\* In the What Changed From 2010 to 2013? section, remove the fragment "Probably because this area is being looked at harder, not because issues are actually more prevalent." I agree with this sentiment, but I think it's hard to substantiate. A: We think it’s OK to express an opinion here.

\* In the What Changed from 2010 to 2013? section, change "but now deserves a category int its own right" to "but now has a category of its own". A: Done

APPLICATION SECURITY RISKS

\* In the What Are Application Security Risks? section change "Similarly, the harm that is caused my range from nothing, all the way through putting you out of business." to "Similarly, the harm caused may be of no consequence or it may put you out of business." A: Done

\* In the What's My Risk? section, change "We chose the name that is best known and will achieve the highest level of awareness" to "We chose names that accurately reflect the risks and, where possible, align with common terminology likely to raise awareness." A: Done

SECURITY RISKS - 2013

\* Remove the bullets for all risks. The visual layout already works fine without them. A: Done.

\* In A1, change "accessing unauthorized data" to "accessing data without proper authorization". A: Done

\* In A2, change "sessions tokens, or" to "or session tokens, or to". A: Done

\* In A5, change "All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date." to "Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date." A: Done

\* In A6, change "tax ids" to "government issued identifiers". A: If there was room, we’d change it, but that box is full so and this change would add another line that doesn’t fit.

\* In A6, transpose "identity theft" and "credit card fraud" because that is the same ordering as for "credit cards" and "government issued identifiers" earlier in this summary. A: Done

\* In A7, change "Virtually all" to "Most". A: Done

\* In A7, change "unauthorized functionality" to "functionality without proper authorization." A: Done

\* In A9, change "Vulnerable components" to "Components". Add a comma after "modules", and change "privilege" to "privileges". A: Done

\* In A9 change "they can cause" to "such components can facilitate". A: Done

Here's Batch 2, which goes up to the end of A6.

OWASP TOP 10 APPLICATION SECURITY RISKS - 2013

\* Give Insecure Direct Object References an acronym, IDOR. I see this used casually, but it's probably time to raise awareness further so people have a common acronym.   A: We’ve never heard of this acronym so we’re hesitant to introduce it to the Top 10.

A1 INJECTION

\* The font weight in the opening table seems to be heavier than for other vulnerabilities. It should probably be made consistent. A: Done – Actually it wasn’t bold for A1 and bold for all others, so we made it bold too.

\* In the Security Weaknesses box, change "SQL, LDAP, or XPath queries, OS Comands, XML parsers, program arguments, etc." to "SQL, LDAP, XPath, or NoSQL queries; OS commands; XML parsers, program arguments, and so forth." A: Done

\* In the Security Weaknesses box, change "but more difficult via testing" to "usually harder to discover via remote testing". A: Done

\* In the Security Weaknesses box, change "find them" to "find injection flaws. A: Done

\* In the Example Attack Scenario section, make the single quote characters a different color than red. A: Done – they are now black.

\* In the Example Attack Scenario section change "in their browser" to "in the browser". A: It is their browser, so no change.

\* In the Example Attack Scenario section, make   ' or '1'='1   be red in the main text. Remove the colon after the word "send". Change "customer's" to "customer's record". A: Done – except for removing the :

\* In the How Do I Prevent Injection? box, change "Preventing injection" to "Prevention of injection" A: Disagree – no change.

\* In the How Do I Prevent Injection? box, change #1's final sentence to say "For APIs that support partial parameterization, such as stored procedures, implement additional data validation." A: Disagree – input validation may not be the fix. Using prepared statements inside of the stored procedure is the better approach usually. We don’t want to specify a solution here because its technology specific.

A2 BROKEN AUTHENTICATION AND SESSION MANAGEMENT

\* In the Security Weakness box, change "etc." to "and so forth". A: No change. That’s a style preference. We prefer etc.

\* In the Am I Vulnerable to Hijacking section, change #1 to "Are credentials stored in plaintext or with poorly designed hashing or encryption schemes?"

\* In the Am I Vulnerable to Hijacking? section, change #5 to "Do session IDs never timeout and is user logout broken or missing?" A: Due to comments from someone else, we changed this to: “Session IDs don’t timeout, or user sessions or authentication tokens, particularly single sign-on  (SSO) tokens, aren’t properly invalidated during logout.” We believe this also addresses your comment.

\* In the Am I Vulnerable to Hijacking? section, change #7 to "Are passwords, session IDs, and other credentials sent over cleartext connections?". A: Agree. Changed to: over unencrypted connections.

\* In Example Attack Scenarios, Scenario #3, change "encrypted" to "properly hashed".   A: Done

A3 CROSS-SITE SCRIPTING (XSS)

\* In the Security Weaknesses box, change "via testing" to "via automated testing". A: Disagree. There are lots of XSS flaws that can’t easily be found with automated testing. Many tools have serious trouble with both Stored and Deferred XSS (i.e., XSS that occurs over multiple pages).

\* In the Technical Impacts box, change "etc." to "and so forth." A: No change – as above.

\* In the Am I Vulnerable to XSS? section, remove the first paragraph. It doesn't explain whether one is vulnerable. The entire paragraph could be reworded to "XSS vulnerabilities usually surface when input data is not thoroughly validated  prior to processing and is not thoroughly escaped prior to rendering. In JavaScript contexts, non-sanitizing API methods are particularly at risk." A: Comments from someone else caused us to rewrite this paragraph like so:

You are vulnerable if you do not ensure that all user supplied input is properly escaped, or you do not verify it to be safe via input validation, before including that input in the output page. Without proper output escaping or validation, such input will be treated as active content in the browser. If AJAX is being used to dynamically update the page, are you using [safe JavaScript APIs](https://www.owasp.org/images/c/c5/Unraveling_some_Mysteries_around_DOM-based_XSS.pdf)? For unsafe JavaScript APIs, encoding or validation must also be used.

\* In the Am I Vulnerable to XSS? section, in the second paragraph, change "which makes automated detection difficult." to "making automated detection difficult." Change "pen testing" to "penetration testing". A: Done

\* In the Am I Vulnerable to XSS? section, change "AJAX" to "Ajax". This is the common way to name it now. A: Done

\* In the Am I Vulnerable to XSS? section, change "much more difficult to detect via automated tools." to "much more difficult to detect via automated tools lacking runtime instrumentation." A: Disagree – this is too specific a comment. Depends on the tool.

\* In the How Do I Prevent XSS? section, change "Preventing XSS requires keeping untrusted data separate from active browser content" to \* "Prevention of XSS requires separation of untrusted data and active browser content." A: Done

\* In the How Do I Prevent XSS? section, in #1 remove the text "The preferred option is to". A: Disagree. This is the primary recommendation of the OWASP XSS Prevention Cheat Sheet.

\* In the How Do I Prevent XSS? section, consider adding a #4 of "Consider use of malice detection techniques defined in the AppSensor project." A: We love AppSensor. But it’s not an XSS specific thing. We’d love to mention it somewhere in the Top 10. If you can find a good spot for a reference/recommendation to use/look at it, please let us know.

\* In the Example Attack Scenario section, make the single quotes a different color. A: Done

\* Consider adding the AppSensor link to References. A: Again, no change per comment above.

A4 INSECURE DIRECT OBJECT REFERENCES

\* Consider creating an acronym of "IDOR" to put this thing on the map. A: We’ve never heard this acronym used before. As such we are hesitant to include it in the Top 10.

\* In the Attack Vectors box, change "to another object the user isn't authorized for" to "the user isn't supposed to access." A: No change. We think the current wording is more accurate/precise.

\* In the Security Weaknesses box, change "such flaws and code analysis" to "such flaws. Code analysis". A: Done

\* In the Technical Impacts box, change "name space is sparse" to "object references are unpredictable". A: Done

\* In the Am I Vulnerable? section, change #1 to say "For direct references to restricted resources, does the application fail to verify the user is authorized to access the exact resource requested?". A: Done

\* In the Am I Vulnerable? section, change #2 to say "If the ference is an indirect reference, does the mapping to the direct reference fail to limit the values authorized for the current user?". A: Done

\* In the Am I Vulnerable? section, change the last sentence to say "Automated tools do not typically perform A/B object reference enumeration and determination tests." A: No change. We think the current language is more clear.

\* In the Example Attack Scenario section, change "their browser" to "the browser. Also, change "whatever account number they want" to "any account number". A: No change. The current text is accurate/equivalent.

\* In the Example Attack Scenario section, change "If not verified" to "If the application doesn't match the current user's account number(s) against the 'acct' parameter". A: We added the word ‘properly’ in front of verified.

\* In the How Do I Prevent This? section, remove the first sentence. Just have the numbered list. That's enough. A: No change. All the other top 10 items have an intro sentence here so for consistency, we’d prefer to leave it here too.

\* In the References section remove the the text and trailing space of "which is ".  A: Done

A5 SECURITY MISCONFIGURATION

\* In the Attack Vectors box, replace "etc." with "and so forth," A: No change, as before.

\* In the Security Weaknesses box, add "database" to the list. A: Done

\* In the Security Weaknesses box, change "network administrators" to "system administrators". A: Done

\* In the Security Weaknesses box, change "etc." to "and so forth". A; No change, as before.

\* In the Business Impacts box, change "All your data" to "All of your data".  A; Done

\* In the Am I Vulnerable to Attack? section, change each numbered item to express the negative case. In other words. A: Done – words might be slightly different but same gist.

1. Is your software (e.g., OS, web/app server, DMS, applications) out of date, not following a frequent patching schedule?

2. Are unnecessary features enabled (e.g., ports, services, pages, accounts, privileges)?

3. Have default accounts not yet been disabled with their passwords rotated?

4. Does your error handling emit stack traces and other overly informative error messages?

5. Are the security settings in your development frameworks (e.g., Struts, Spring, [ASP.NET](http://ASP.NET)) not yet set to secure values?

\* In the Am I Vulnerable to Attack? section, change the last sentence to "Without a concerted, repeatable configuration process, system architecture is at a higher risk." A: Not sure how architecture is affected by this risk. The system itself is at higher risk. Is that what you mean to say here?

\* In Example Attack Scenarios, Scenario #2 change "reverses" to "decompiles and reverse engineers". Change "She" to "Attacker". A: 1st part done. We need to use a pronoun in this paragraph already, so we think its fine to leave she in the paragraph. Otherwise we’d have to use he instead.

\* In the How Do I Prevent This? item #1 add a parenthetical after "identically" that says "(n.b., with different passwords used in each environment)". A: Done

\* In the How Do I Prevent This, item #3, change "good separation and security between" to "effective secure separation of". A: Done

A6 SENSITIVE DATA EXPOSURE

\* Change the Attack Vectors box to say "Attackers typically steal encryption keys, perform man-in-the-middle attacks, copy server files, or examine browser caches and history." A: We think the intro sentence is important. We did tweak it now say: Attackers typically don’t break crypto directly. They break something else, such as steal keys, do man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user’s browser.

\* In the Security Weaknesses box, change "particularly weak hashing solutions to protect passwords." to "particularly with weak password hashing techniques." A: Done

\* In the Security Weaknesses box, change "but hard to exploit" to "but harder to massively exploit." A: added ‘on a large scale’ to end of sentence instead.

\* In the Security Weaknesses box, change "External attackers have difficulty detecting most of these types of flaws due to limited access and they are also usually harder to exploit." to "External attackers usually have more difficulty detecting remote server flaws due to limited access and data that looks opaque." A: Not enough room to add all this. Did change the sentence to focus on server side flaws though.

\* In the Technical Impacts box, change add a comma after "Typically", and replace "etc." with "and so forth.". A” Comma added.

\* In the Am I Vulnerable to Data Exposure section, again, change from normative to negative...  A: Done – Although not using exactly these suggested changes.

For all such data:

1. Is data that will be stored long term insecurely stored in plaintext?

2. Is data sent across untrusted networks (e.g., the Internet) sent in cleartext (e.g., HTTP, not HTTPS)?

3. Are broken cryptographic algorithms in use?

4. Is there no cryptographic key management?

5. Do Cache-Control and Expires headers incorrectly permit caching in intermediate caches or for longer than necessary?

\* In the Example Attack Scenarios section, Scenario #2, change "all their" to "the user's". A: Done

\* In the Example Attack Scenarios section, Scenario #3, change "file upload" to "IDOR". Change "All the" to "All of the". A: ‘of’ added. IDOR not used since not introduced earlier.

\* In the How Do I Prevent This? item #2, remove the sentence "Data you don't have can't be stolen." Alternatively, reward to "Data you've digitally discarded is harder to steal."  A: We feel it’s important to leave in. There isn’t sufficient room to add more clarity. We’d like to add the word ‘anymore’ after ‘don’t’ but even that won’t fit.

\* In the How Do I Prevent This? item #4, add a final sentence of "Store master salts on systems independent of password hashes." A: This is a pretty specific implementation detail we’d rather not specify.

Third and final installment below for A7 and beyond.  
  
Looking back through suggested edits, I think the important thing for the "Am I Vulnerable to ...?" sections is a standardized way of defining vulnerability. I hope by rewording it becomes clearer what the antipatterns are that clue someone into whether their application is at risk.  A: We received similar feedback fron Torsten Gigler too. We think they are all much better now at focusing on antipatterns to be consistent with the title of the section. Please check out the current latest update (attached).

Regarding the replacement of "etc." with "and so forth" that's more of a style thing than anything else. Not a biggie, just the inner Strunk & White emerging. I get that it saves space to use "etc.". We are just used to etc., and it is indeed shorter, so we’ve left them alone.

A7 MISSING FUNCTION LEVEL ACCESS CONTROL  
\* In the Am I Vulnerable to Forced Access? section change #2 to "Are authentication and authorization unchecked?" and change #3 to "Do checks rely solely upon information provided by the attacker?" A: Done – with similar words.  
\* In the Am I Vulnerable to Forced Access? section, change "Some proxies support this type of analysis." to "Compare the server responses. If they are alike, you're probably vulnerable." A: We were able to squeeze both points in.  
\* In the last sentence of the Am I Vulnerable to Forced Access? section, change "Automated" to "Untuned automated". A: We think saying unlikely is sufficient.  
\* In the How Do I Prevent Forced Access? section, change "all your" to "all of your". A: Done  
  
A8 CROSS-SITE REQUEST FORGERY (CSRF)  
\* Change the Impact to "Attackers can cause victims to change any data the victim is allowed to change or perform any other function the victim is authorized to use. This may include login forms and logout." A: We were able to split this into 2 sentences, but with slightly different wording.  
\* In the Am I Vulnerable to CSRF? section, change "don't count since this information is also included in forged requests." to "don't count as a CSRF because this information is already included in forged requests." A: A similar change was implemented.  
\* In the Example Attack Scenario section, change "their" to "the attacker's", and remove the phrase "like so". A: Done  
\* In the How Do I Prevent CSRF? section, change "Preventing CSRF" to "Prevention of CSRF". A: We prefer the current wording.  
\* In How Do I Prevent CSRF? #1, change "which is subject" to "which is more prone".  A: Done  
\* In How Do I Prevent CSRF? #2, change "runs the risk" to "runs a greater risk". A: Done  
\* In the How Do I Prevent CSRF? section, change "includes CSRF methods" to "includes anti-CSRF methods". Or just drop the word "CSRF" here. A: Done  
  
A9 USING COMPONENTS WITH KNOWN VULNERABILITIES   
\* In the Attack Vectors box, change "They customize the exploit as needed and execute the attack." to "Attacker customizes the exploit as necessary." A: Not sure this is any clearer. No change.  
\* In the Technical Impacts box, change "etc." to "and so forth."  A: No change, as before.  
\* In the Technical Impacts box, change "The impact could be minimal, up to complete host takeover and data compromise." to "The impact could range from minimal to complete host takeover and data compromise.". A: Done

\* In the Example Attack Scenarios section, change "from the trivial" to "ranging from the trivial". A: Done  
\* In the How to Prevent This? section, change "up-to-date" to "up to date" to match the rest of the document. A: Done  
\* In the How Do I Prevent This? section, change "Many open source" to "Many". Backported security patches are rare from just about everyone, unfortunately. A: We changed ‘open source projects’, to say ‘component projects’  
  
A10 UNVALIDATED REDIRECTS AND FORWARDS   
\* In the Attack Vectors box, change "since" to "because". Also, change "Attacker targets" to "Or, attacker targets". A: We can’t make either of these changes fit in the box, so we left it alone.  
\* In the Security Weaknesses box, change "since" to "because". A: Done  
\* In the Am I Vulnerable to Redirection? section change #1's final sentence to say "Is the target URL not validated against a whitelist?" A: Done  
\* In the Example Attack Scenarios section, change "and then forwards the attacker to an administrative function that she would not normally be able to access." to "and then forward (invoke through code inclusion) administrative functionality for which the attacker isn't authorized."  A: We changed this to match the last part of your suggestion, but didn’t include the parenthetical piece.  
\* In the How Do I Prevent This? section #3, change the verbiage to "Use a keyed map of valid identifiers and their corresponding URLs." A: We know what we have there is a bit wordy, but its more explanatory than what you suggest, and since we have the room we’d prefer to leave it in.  
  
WHAT'S NEXT FOR DEVELOPERS  
\* Remove the bullet characters. The layout already suffices. A: We’d like to, if we can figure out how.  
\* Change "Providing developers with a set of standard security controls" to "A set of standard security controls". A: Done  
  
WHAT'S NEXT FOR VERIFIERS  
\* Change "security code review" to "secure code review". A: Done  
\* Update the Live CD verbiage to refer to the latest terminology, and refer to it in its virtual machine form as well, as that's becoming more common than use of an actual CD. A: Done  
\* Change "not providing the security itself" to "not providing all of the security itself". A: Done  
\* Change "security analysts" to "security analysts and developers", and "analyst" to "analyst and developer". Developers love proxies! A: Done – although with slightly different wording.  
\* Change "its FREE" to "it's FREE". A: Done  
  
WHAT'S NEXT FOR ORGANIZATIONS  
\* Change "no longer a choice" to "no longer optional". A: Done  
\* Change "Pen Testing" to "Penetration Testing". A: Done  
\* Change "Remediation, etc." to " and Remediation." A: Done  
\* Change "application coverage, etc." to "application coverage, and defect density by type and instance counts." A: Done  
  
NOTE ABOUT RISKS  
\* Change "system owner can" to "system owners can be". A: Done  
\* Change "We don't know how important your applications and data are" to "You are best equipped to judge the importance of your application and data". A: Done  
\* Change "3" to "three". A: Done  
\* Add a reference to the published vulnerability data. A: Done  
\* Change "is willing to accept" to "is willing to accept within in its organizational culture, industry, and regulatory environment". A: Done – with slightly different wording.  
  
DETAILS ABOUT RISK FACTORS  
\* Change "Insecure DOR" to "IDOR". A: No change, since not introduced before.  
\* I sampled the Exploitability column, and it seems accurate. A full pass by a different QC person for the other columns would be good.  
\* Change "Other important application security risks" to "Just a small sampling of important application security risks" A: There isn’t enough room. The word include at the end is there to indicate this isn’t an exhaustive list by any means.

1. Adam Baso and the project exchanged several additional emails and more minor updates were made based on him reviewing the updates we made based on his original set of comments.

Using the latest PDF you sent and the commentary from this email, I recognized several places we can tighten up the meaning or grammar.

ABOUT OWASP  
\* In the Forward column change "everything in a process model" to "everything prescribed by some process model". The current sentence is slightly ambiguous in that one may take it to mean not to use a process model, when the point - as amplified in the following sentence - is to define something tailored to the organization. A: Done

\* In the Forward column change "existing organization's" to "organization's existing". A: Done

INTRODUCTION

\* In the Warnings column change "attacks methods refined" to "attack methods are refined".  A: Done

\* In the Acknowledgments section change "and to its primary authors" to "and to primary authors".  A: We prefer to leave its in there.

\* In the Acknowledgments section change "those that" to "those who". A: Done

RELEASE NOTES

\* In the What Changed from 2010 to 2013? section, change "Probably because" to "We believe this is probably because". This turns the fragment into a full sentence. A: Done

APPLICATION SECURITY RISKS

\* In the What Are Application Security Risks section change "be on no consequence" to "be of no consequence". A: Done – Good catch!

OWASP TOP 10 APPLICATION SECURITY RISKS - 2013

\* In A5, add a comma after the word "maintained" to insert a mental pause to show the opposition between good practice and what is the unfortunate reality. A: Done

\* In A6, consider changing "ids" to "IDs". I believe this form is more customary. A: Done

A1 INJECTION

\* In the How do I Prevent Injection? box, change "Be careful of APIs" to "Be careful with APIs". A: Done

I agree with you on parameterizing from *within* stored procs if possible, by the way. I now understand why you took the more conservative approach with the verbiage on stored procs.

A3 CROSS-SITE SCRIPTING (XSS)

I agree with you that *automated* testing, particularly without tuning for canary values for stored or deferred values, won't make XSS flaw discovery "fairly easy". Best to keep the verbiage generic so as to stave off people thinking they can "just" run a scanning utility.

I now understand why you want to keep "The preferred option is to": for emphasis on this specific technique. I want to clarify here that my point was to remove what I thought was unnecessary text, not to question the technique as being of utmost importance! It's fine as is.

A5 SECURITY MISCONFIGURATION

\* In the Am I Vulnerable to Attack? box, in #2, add a comma after "e.g.". A: Done (and you missed one :-)  - I fixed the other too.

\* For the recommended verbiage for the final sentence of the Am I Vulnerable to Attack? box, you're right, the word "architecture" is unnecessary. I recommend the following instead:

"Without a concerted, repeatable application security configuration process, systems are at a higher risk."  A: Done

A6 SENSITIVE DATA EXPOSURE

\* Add a period at the end of the final sentence of the Am I Vulnerable to Data Exposure? box.  A: Done – picky picky :-)

And finally...after re-reading the document once more:

A1 INJECTION

\* Change "e.g., Hibernate Query Language (HQL)" to "(e.g., Hibernate Query Language [HQL])" or, if you're really pedantic, "(e.g., Hibernate Query Language (HQL))".  A: Done

\* Change "their browser" to "her browser".  A: Done

A4 INSECURE DIRECT OBJECT REFERENCES

\* Change "their browser" to "his browser".  A: Done – and another somewhere else.

A5 SECURITY MISCONFIGURATION

\* Add a comma after "e.g." in the Am I Vulnerable to Attack? box's #2. – A: already done 

A6 SENSITIVE DATA EXPOSURE

\* In the How Do I Prevent This? box, #5, change "pages displaying" to "pages containing".  A: Done

A8 CROSS-SITE REQUEST FORGERY (CSRF)

\* In the Am I Vulnerable to CSRF? box, change "since" to "because".  A: Done

A9 USING COMPONENTS WITH KNOWN VULNERABILITIES

\* In the Attack Vectors box, change "They customize" to "He customizes". I believe this rounds out the female/male ratios for attack scenarios.  A: Done

\* In the Technical Impacts box, change "The impact could be minimal, up to complete host takeover and data compromise." to "The impact could range from minimal to complete host takeover and data compromise.".  A: Done

INTRODUCTION

\* Change "8 firms" to "7 firms" and "4 tool vendors (2 static and 2 dynamic) " to "3 tool or SaaS vendors (covering 2 static and 2 dynamic analysis offerings)". I believe this will treat both Jeremiah's and Christian's comments on this matter. Jeremiah's comment, but with a small modification from 8 to 7, may actually make this simpler: "The OWASP Top 10 is based on risk data from 7 firms that specialize in application security." A: Done – It now says: The OWASP Top 10 for 2013 is based on 8 datasets from 7 firms that specialize in application security, including 4 consulting companies and 3 tool/SaaS vendors (1 static, 1 dynamic, and 1 with both).

1. Stefan Finkenzeller provided this comment (through Torsten):

Do you think SMTP-Header-Injection is important enough to be listed in A1 as an additional example for Injection? (Perhaps in the 'Security Weakness' box)

This type of Injection could be caused by Web-Mail input forms.

I googled some older articles about this, like http://www.cakesolutions.net/teamblogs/2008/05/08/email-header-injection-security/

I did find only 1 article from OWASP ( https://www.owasp.org/index.php/Testing\_for\_IMAP/SMTP\_Injection\_%28OWASP-DV-011%29 ) that targets a slightly different impact.

We updated the weakness box to say: “They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc.”