OWASP

OWASP New Zealand Day 2013

Bad Smells That Lead to Bad Security
aka Developing your Security Spidey Senses

Andy Prow – Aura InfoSec
Kirk Jackson - Xero

xero

Intro

Andy Prow

Managing Director
Aura Information Security Ltd

Kirk Jackson

Security Officer
Xero


epSos.de
(cc) BY
By epSos.de

Spidey Senses

"A vague but strong sense of something being wrong, dangerous or suspicious"

- Should trigger almost immediately.
- Suggests there is something to be wary of or something that needs investigating
- Good hackers have great ones.
- If smart users have one, they run…

Bad smells

- Unlocked Vault – Fishy
- Too Trusting – Sickly sweet
- Spilling your secrets – Vomit
- Back door is open – Garbage
- DIY – Oily rag
- Dodgy foundations – Sewers
- Risky Business - Fear

DEVELOP YOUR SECURITY SPIDEY-SENSE

aura INFORMATION SECURITY

Bad smell	Examples	#
Unlocked vault *Smell: Fishy* Walk past and see the treasure	Site mixes HTTP and HTTPS	A6
	Length and character limits for passwords	A2
	Missing security HTTP headers	A6
Too trusting *Smell: Sickly sweet* No-one questions or challenges you	Files that expose content (robots.txt, sitemap.xml, VCS etc)	
	Cached search engine results	
	Entering 10 invalid passwords, can still log in	A6
	Change password without entering existing password	
	Certain characters give errors, or display incorrectly: < ' " ;	
Spilling your secrets *Smell: Vomit* You can see stuff you shouldn't be able to see	Client side validation of user input	A2
	User content displays directly on screen (e.g. reflected XSS)	A2
	Request contains SQL "select x from y where z"	A3
	Signup or password reset email contains a password	A3
	Change password form shows current password	A3
	Site lists characters that are banned in text fields	A1
	Customer id in the URL	A6
Back door is open *Smell: Trash* Look around the other side of the building	A number in the URL that increases by one	A6
	You can tell the underlying technology (e.g. LDAP, SQL, XML)	
	Sensitive data is unencrypted	A4
	Authorisation - by URL / menu	
	Admin site available to the world	
D.I.Y. *Smell: Oily Rag* Invented something themselves	Production data in a test environment	A6
	Doesn't use out of the box forms authentication	A7
	Custom single-sign-on	A7
	Homebuilt encryption	
Dodgy foundations *Smell: Sewer* Built or extended over crumbling foundations	Cookie flags, scope (domain + path), expiration	A2
	Multiple apps, including third party	A2
	A mix of technologies (e.g. PHP + ASP.NET)	
	Cross-domain javascript / CSS	A2
	Rich Client component (e.g. Flash, Silverlight, JS)	
	Old OS, server, ASP.NET version	
Risky Business *Smell: Fear* Risky things that you should be wary of	Built using Webforms	
	Writing directly to disk	
	Application accepts file uploads	A5
	Parsing XML, HTML or other formats	
	Credit card payments	
	Mobile app talking to API	A4

See the OWASP Top 10 2013 - owasp.org

An electronic version of this sheet is available at: www.aurainfosec.com/publications
View our other talks at hack-ed.com

xero

The Unlocked Vault

Something Smells Fishy…

Our vault contains rotten fish (and heaps of cash) but I shouldn’t be able to smell it outside...

Senses tell me:

Basic security steps are missing entirely.

Examples

- Site mixes HTTP and HTTPS
- Length and character limits for passwords
- Missing security HTTP headers
- Files that expose content (robots.txt, sitemap.xml, VCS etc)
- Cached search engine results

Too Trusting

Something Smells Too Sickly Sweet…

My security guard is way too smiley and seems to think his job is let everyone in…

Senses tell me:

Information and actions going into the system are being trusted too openly.

Examples

- Entering [10] invalid passwords, can still log in
- Change password without entering existing password
- Certain characters give errors, or display incorrectly:
  < ' " ; ☺
- Client side validation of user input
- User content displays directly on screen (e.g. reflected XSS)
- Request contains SQL "select x from y where z”
- No random tokens in the form data (CSRF)
- URLs are in the query string

Spilling your secrets

Something Smells of Vomit…

The security guard has spilt his guts…

Senses tell me:
Information is being leaked out of the system.

Examples

- Signup or password reset email contains a password
- Change password form shows current password
- Site lists characters that are banned in text fields
- Customer id (or similar object reference) in the URL
- A number in the URL that increases by one (or is an obvious format)
- You can tell the underlying technology (e.g. LDAP, SQL, XML)

Something Smells of Garbage…

I’ve been to the back of the bank, and the security stinks…

Senses tell me:

Front-door is good but someone’s left alternative avenues into the system.

Examples

- Sensitive data is unencrypted
- Authorisation - by URL or menu
- Admin site available to the world
- Production data in a test environment

Something Smells of Oily Rags…

This does not smell like a professional job…

Senses tell me:

Some core security features are not using proven techniques but have been home-grown.

Examples

- Doesn't use out of the box authentication
- Custom single-sign-on
- Homebuilt encryption
- Cookie flags, scope (domain + path), expiration

Something Smells of Sewers…

The bank looks great, but it’s built on a crumbling sewer.

Senses tell me:

There is a poor underlying architecture or technology beneath this system.

Examples

- Multiple apps, including third party
- A mix of technologies (e.g. PHP + ASP.NET)
- Cross-domain javascript / CSS
- Rich Client component
  (e.g. Flash, Silverlight, JS)
- Old OS, server, ASP.NET, PHP, Java version
- Writing directly to disk

Risky Business

Something Smells of Fear…

Smells like someone's taken a risk and implemented something dangerous.

Senses tell me:

Features requiring strong security have not been implemented securely enough.

Examples

- Application accepts file uploads
- Parsing XML, HTML or other file formats
- Credit card payments
- Mobile app talking to API

Billion Laughs

&m;

Billion Laughs

&l;&l;&l;&l;&l;&l;&l;&l;

Billion Laughs

&k;&k;&k;&k;&k;&k;&k;&k; &k;&k;&k;&k;&k;&k;&k;&k;
&k;&k;&k;&k;&k;&k;&k;&k; &k;&k;&k;&k;&k;&k;&k;&k;
&k;&k;&k;&k;&k;&k;&k;&k; &k;&k;&k;&k;&k;&k;&k;&k;
&k;&k;&k;&k;&k;&k;&k;&k; &k;&k;&k;&k;&k;&k;&k;&k;

Billion Laughs

&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;

Billion Laughs

&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;& &j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j; **Boom!** j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;& &j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;
&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&
j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;&j;

Bad smells

- Unlocked Vault – Fishy
- Too Trusting – Sickly sweet
- Spilling your secrets – Vomit
- Back door is open – Garbage
- DIY – Oily rag
- Dodgy foundations – Sewers
- Risky Business - Fear

DEVELOP YOUR SECURITY SPIDEY-SENSE

aura INFORMATION SECURITY

Bad smell	Examples	#
Unlocked vault *Smell: Fishy* Walk past and see the treasure	Site mixes HTTP and HTTPS	A6
	Length and character limits for passwords	A2
	Missing security HTTP headers	A6
Too trusting *Smell: Sickly sweet* No-one questions or challenges you	Files that expose content (robots.txt, sitemap.xml, VCS etc)	
	Cached search engine results	A2
	Entering 10 invalid passwords, can still log in	A3
	Change password without entering existing password	A3
	Certain characters give errors, or display incorrectly: < ' " ;	A3
Spilling your secrets *Smell: Vomit* You can see stuff you shouldn't be able to see	Client side validation of user input	A1
	User content displays directly on screen (e.g. reflected XSS)	A6
	Request contains SQL: "select x from y where z"	A6
	Signup or password reset email contains a password	
	Change password form shows current password	A4
	Site lists characters that are banned in text fields	
Back door is open *Smell: Trash* Look around the other side of the building	Customer id in the URL	A6
	A number in the URL that increases by one	A7
	You can tell the underlying technology (e.g. LDAP, SQL, XML)	A7
	Sensitive data is unencrypted	
	Authorisation - by URL / menu	A2
	Admin site available to the world	A2
D.I.Y. *Smell: Oily Rag* Invented something themselves	Production data in a test environment	A2
	Doesn't use out of the box forms authentication	
	Custom single-sign-on	
	Homebuilt encryption	
	Cookie flags, scope (domain + path), expiration	
Dodgy foundations *Smell: Sewer* Built or extended over crumbling foundations	Multiple apps, including third party	A5
	A mix of technologies (e.g. PHP + ASP.NET)	A4
	Cross-domain javascript / CSS	
	Rich Client component (e.g. Flash, Silverlight, JS)	A6
	Old OS, server, ASP.NET version	A7
	Built using Webforms	
Risky Business *Smell: Fear* Risky things that you should be wary of	Writing directly to disk	
	Application accepts file uploads	
	Parsing XML, HTML or other formats	
	Credit card payments	
	Mobile app talking to API	

See the OWASP Top 10 2013 - owasp.org

An electronic version of this sheet is available at: www.aurainfosec.com/publications
View our other talks at hack-ed.com

xero

OWASP Top 10 – updated for 2013

A1-Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2-Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
A3-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A4-Insecure Direct Object References	A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.
A5-Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.
A6-Sensitive Data Exposure	Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.
A7-Missing Function Level Access Control	Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.
A8-Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.
A9-Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.
A10-Unvalidated Redirects and Forwards	Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

Summary

- Develop your own Spidey-Senses
- OWASP Top 10 2013 owasp.org
- Publications on www.aurainfosec.com
- View our other talks at hack-ed.com

DEVELOP YOUR SECURITY SPIDEY-SENSE

aura INFORMATION SECURITY

Bad smell	Examples	#
Unlocked vault *Smell: Fishy* Walk past and see the treasure	Site mixes HTTP and HTTPS	A6
	Length and character limits for passwords	A2
	Missing security HTTP headers	A6
Too trusting *Smell: Sickly sweet* No-one questions or challenges you	Files that expose content (robots.txt, sitemap.xml, VCS etc)	
	Cached search engine results	
	Entering 10 invalid passwords, can still log in	A2
	Change password without entering existing password	A2
	Certain characters give errors, or display incorrectly: < ' " ;	A3
Spilling your secrets *Smell: Vomit* You can see stuff you shouldn't be able to see	Client side validation of user input	A3
	User content displays directly on screen (e.g. reflected XSS)	A3
	Request contains SQL: "select x from y where z"	A1
	Signup or password reset email contains a password	A6
	Change password form shows current password	A6
	Site lists characters that are banned in text fields	
Back door is open *Smell: Trash* Look around the other side of the building	Customer id in the URL	A4
	A number in the URL that increases by one	
	You can tell the underlying technology (e.g. LDAP, SQL, XML)	A6
	Sensitive data is unencrypted	A7
	Authorisation - by URL / menu	A7
	Admin site available to the world	A2
D.I.Y. *Smell: Oily Rag* Invented something themselves	Production data in a test environment	A2
	Doesn't use out of the box forms authentication	
	Custom single-sign-on	A5
	Homebuilt encryption	A4
Dodgy foundations *Smell: Sewer* Built or extended over crumbling foundations	Cookie flags, scope (domain + path), expiration	A6
	Multiple apps, including third party	A7
	A mix of technologies (e.g. PHP + ASP.NET)	
	Cross-domain javascript / CSS	
	Rich Client component (e.g. Flash, Silverlight, JS)	
	Old OS, server, ASP.NET version	
	Built using Webforms	
Risky Business *Smell: Fear* Risky things that you should be wary of	Writing directly to disk	
	Application accepts file uploads	
	Parsing XML, HTML or other formats	
	Credit card payments	
	Mobile app talking to API	

See the OWASP Top 10 2013 - owasp.org

An electronic version of this sheet is available at: www.aurainfosec.com/publications
View our other talks at hack-ed.com

xero

Q&A...