OWASP

Open Web Application Security Project

OWASP TOP 10

2017 Release

Andy Willingham June 12, 2018
OWASP Cincinnati

Agenda

- A quick history lesson
- The Top 10(s)
- Web
- Mobile
- Privacy
- Protective Controls

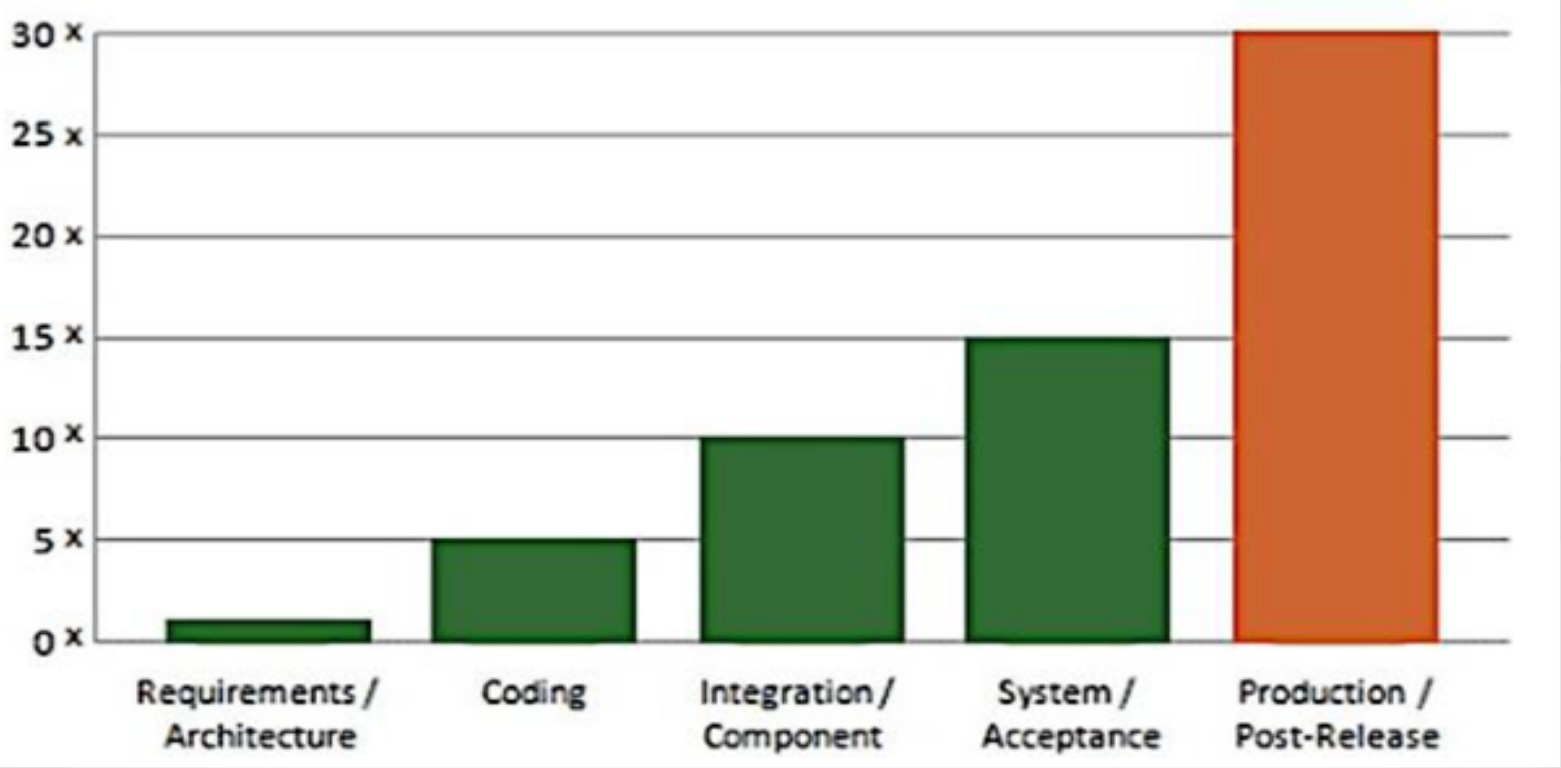
Why have a Top 10?

- Software runs the world (infrastructure, cars, voting, toaster)
- Developers are not perfect
- Developers are not security experts
- Software is complex and has to interact with other systems, apps, and business processes
- Open Source Software is not a panacea
- Software wants to be free (reuse)
- Software is a way in to other systems

Why Software Security Matters

- Cost
  - The cost to fix a found, unexploited security vulnerability far outweighs the cost to prevent it.
  - The cost of a successful exploit of the vulnerability increases by orders of magnitude
  - The cost of lost time that could be spent writing new code over rewriting old code
  - Brand and Reputational cost can decrease marketshare.
- Increases Time to Market when done right
  - Not for initial roll out but for new versions
- Quicker testing
  - Testing smaller chunks of code more often and more throughly
- Quicker fixes
  - Mitigates vulnerabilities faster (think of how long it used to take from discovery to release fix)
  - Quicker fixes improves brand image (responsiveness, takes security seriously, cares about ME)

By The Numbers

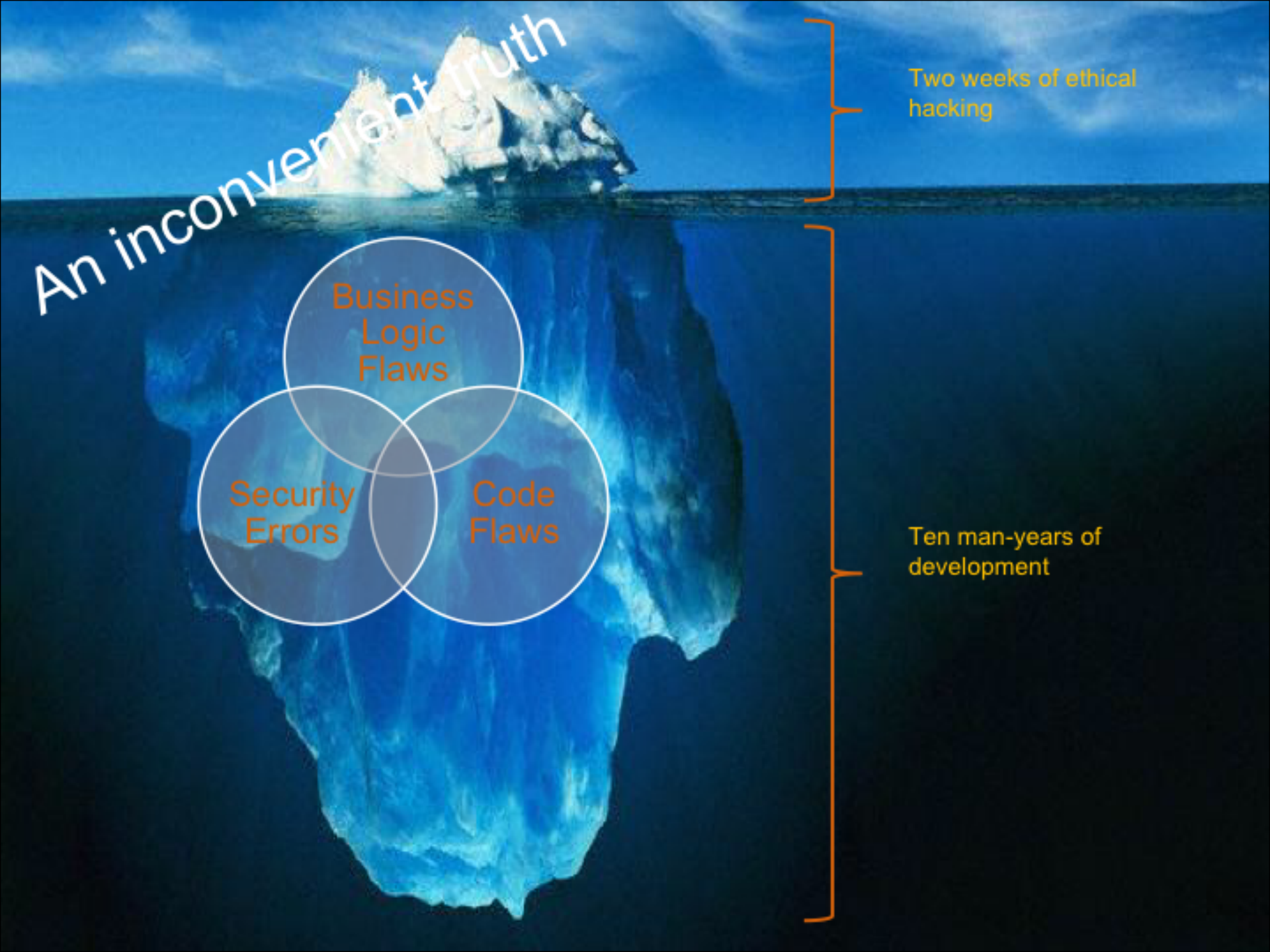


Time and Cost to Fix

www.securityinnovationeurope.com/the-business-case-for-security-in-the-software-development-lifecycle-sdlc

Getting it fixed before it breaks

- Vulnerabilities are introduced in code for a myriad of reasons
  - Rush to market (which can/do lead to below being more common)
  - Code Reuse (typically OSS)
    - One study found that 77% of scanned IOT apps contained OSS and had an ave of 677 vulns per app
    - Many vulns are years old and are high risk
    - Those responsible for remediation are taking longer to fix, if at all
  - Lack of developer secure coding training
  - Lack of QA (testing, peer review)
- You MUST start early
  - Requirements/Design/Architecture
- You MUST verify often
  - QA/Security Testing/Peer Review
- You MUST invest in your teams by providing access to training


An inconvenient truth
Two weeks of ethical hacking
Business Logic Flaws
Security Errors
Code Flaws
Ten man-years of development

OWASP Top 10(s)

- OWASP Top 10 Web Risks
- OWASP Top 10 Privacy Risks
- OWASP Top 10 Mobile Risks
- OWASP Top 10 Proactive Controls

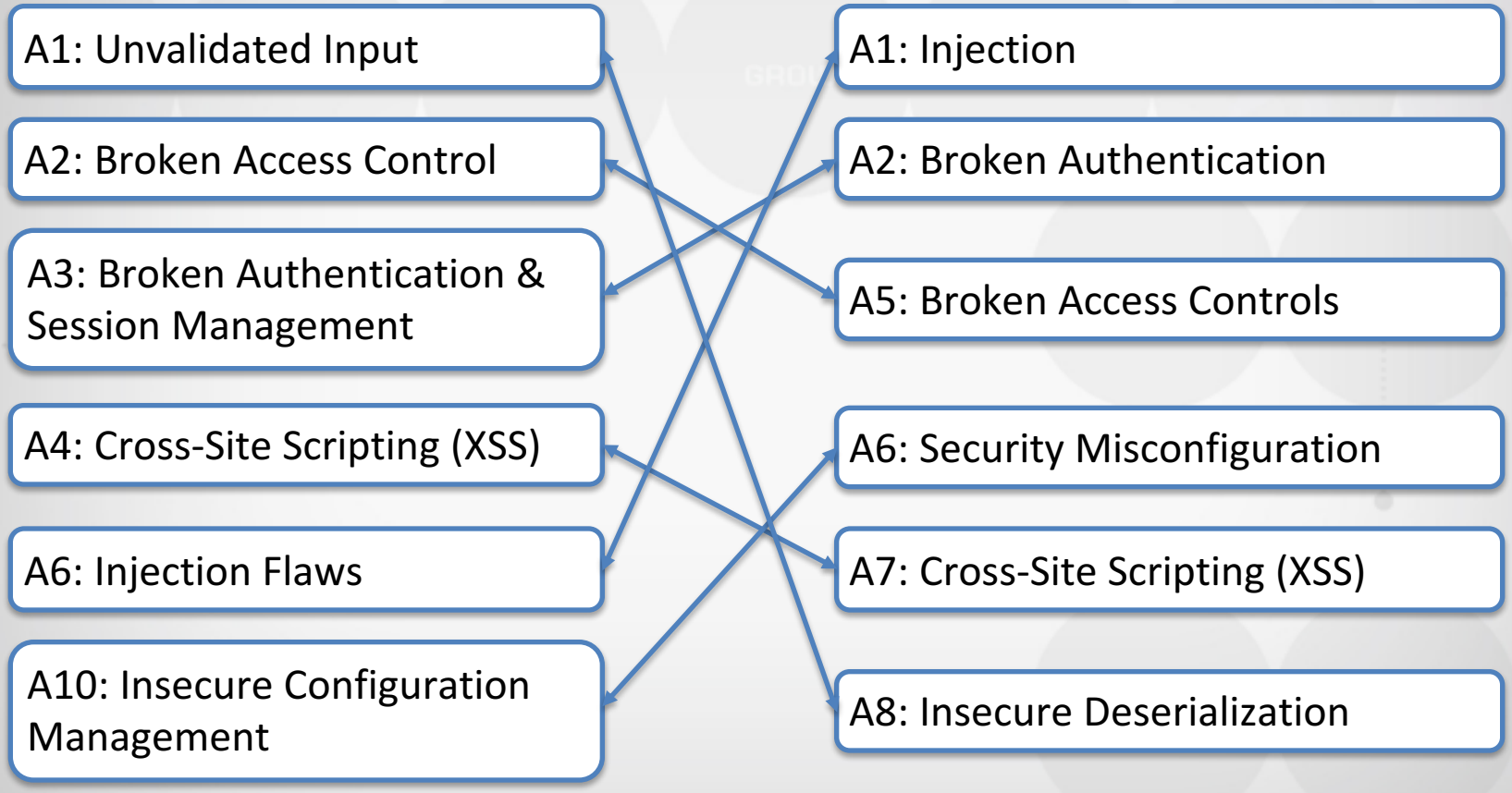
Top 10 Web Risks 2004

- A1: Unvalidated Input
- A2: Broken Access Control
- A3: Broken Authentication and Session Management
- A4: Cross-Site Scripting (XSS) Flaws
- A5: Buffer Overflows
- A6: Injection Flaws
- A7: Improper Error Handling
- A8: Insecure Storage
- A9: Denial of Service
- A10: Insecure Configuration Management

Top 10 Web Risks 2017

- A1: Injection
- A2: Broken Authentication
- A3: Sensitive Data Exposure
- A4: XML External Entities
- A5: Broken Access Control
- A6: Security Misconfiguration
- A7: Cross-Site Scripting (XSS)
- A8: Insecure Deserialization
- A9: Using Components with Known Vulnerabilities
- A10: Insufficient Logging and Monitoring

What's Old is New



Changes to 2017

New issues, supported by data:

- A4:2017-XML External Entities (XXE) is a new category primarily supported by source code analysis security testing tools (SAST) data sets.

New issues, supported by the community:

We asked the community to provide insight into two forward looking weakness categories. After over 500 peer submissions, and removing issues that were already supported by data (such as Sensitive Data Exposure and XXE), the two new issues are:

- A8:2017-Insecure Deserialization, which permits remote code execution or sensitive object manipulation on affected platforms.
- A10:2017-Insufficient Logging and Monitoring, the lack of which can prevent or significantly delay malicious activity and breach detection, incident response, and digital forensics.

Merged or retired, but not forgotten:

- **A4-Insecure Direct Object References** and **A7-Missing Function Level Access Control** merged into A5:2017-Broken Access Control.
- **A8-Cross-Site Request Forgery (CSRF),** as many frameworks include CSRF defenses, it was found in only 5% of applications.
- **A10-Unvalidated Redirects and Forwards**, while found in approximately 8% of applications, it was edged out overall by XXE.

2013 vs 2017

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Top 10: A1 – A5

A1: Injection	Injection flaws, such as SQL, NoSQL, 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	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
A3: Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
A4: XML External Entities (XEE)	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
A5: Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

Top 10: A6 – A10

A6: Security Misconfiguration	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.
A7: Cross-Site Scripting	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. 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.
A8: Insecure Deserialization	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
A9: Using Components Known Vulns	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
A10: Insufficient Logging and Monitoring	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Top 10 Proactive Controls (2018)

1. Define Security Requirements (1?, New)
2. Leverage Security Frameworks and Libraries (9)
3. Secure Database Access (6?, New)
4. Encode and Escape Data (3, partially)
5. Validate All Inputs (4)
6. Implement Digital Identity (5, sort of)
7. Enforce Access Controls (6?)
8. Protect Data Everywhere (7, expanded)
9. Implement Security Logging and Monitoring (8)
10. Handle All Errors and Exceptions (10)

Top 10 Proactive Controls (2016)

1. Verify for Security Early and Often
2. Parameterize Queries
3. Encode Data
4. Validate All Inputs
5. Implement Identity and Authentication Controls
6. Implement Appropriate Access Controls
7. Protect Data
8. Implement Logging and Intrusion Detection
9. Leverage Security Frameworks and Libraries
10. Error and Exception Handling

Top 10 Mobile Risks (2016)

- Improper Platform Usage
- Insecure Data Storage
- Insecure Communication
- Insecure Authentication
- Insufficient Cryptography
- Insecure Authorization
- Client Code Quality
- Code Tampering
- Reverse Engineering
- Extraneous Functionality

Top 10 Privacy Risks (2014)

- Web App Vulnerabilities
- Operator sided Data Leakage
- Insufficient Data Breach Response
- Insufficient Deletion of Personal Data
- Non-Transparent Policy, Terms & Conditions
- Collection of data not required for the primary purpose
- Sharing of data with third party
- Outdated personal data
- Missing or Insufficient Session Expiration
- Insecure Data Transfer

Questions

Apendix

- https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf
- https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10
- https://www.owasp.org/index.php/OWASP_Top_10_Privacy_Risks_Project#tab=Top_10_Privacy_Risks_2
- https://www.owasp.org/index.php/OWASP_Proactive_Controls
- https://www.slideshare.net/mobile/hacker0x01/owasp-top-10-2017-84029876?from_action=save
- www.securityinnovationeurope.com/the-business-case-for-security-in-the-software-development-lifecycle-sdlc
- https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2018-ossra.pdf