OWASP
Open Web Application Security Project

Top 10 Privacy Risks Project

Update & countermeasures

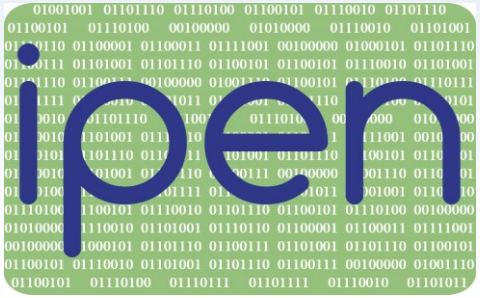
9 September 2016, IPEN Workshop, Frankfurt

Florian Stahl (Project Leader, msg systems, Munich)

Sponsored by msg

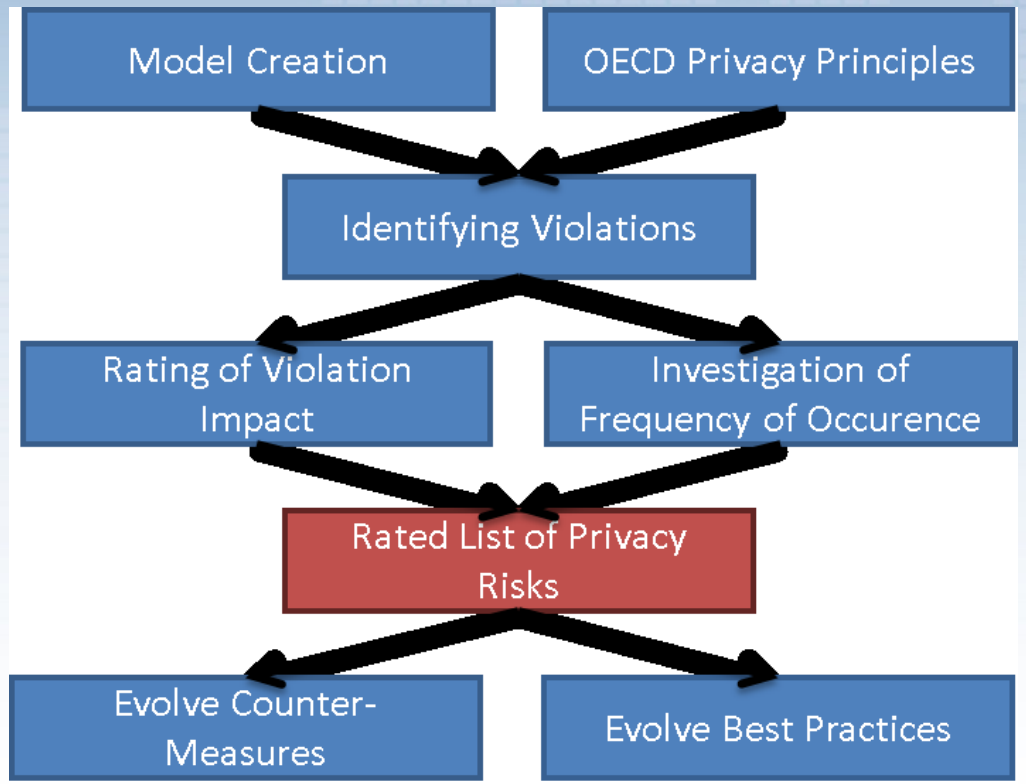
A Look back

- Open source project founded in 2014
- Goal: Educate developers, business architects and legal in web application privacy by showing (technical and organizational) risks
- Best practices, but not always 100% bullet-proof privacy
- Initial presentation of results at IPEN workshop 2014 in Berlin



OWASP
Open Web Application Security Project

Project Method



Results: Top 10 Privacy Risks

P1 Web Application Vulnerabilities

P2 Operator-sided Data Leakage

P3 Insufficient Data Breach Response

P4 Insufficient Deletion of personal data

P5 Non-transparent Policies, Terms and Conditions

P6 Collection of data not required for the primary purpose

P7 Sharing of data with third party

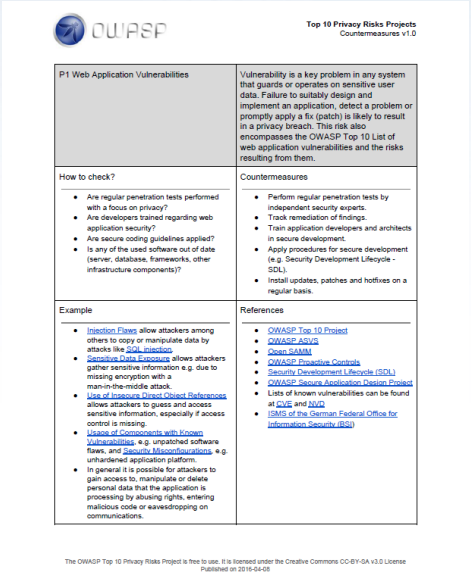
P8 Outdated personal data

P9 Missing or Insufficient Session Expiration

P10 Insecure Data Transfer

Recent developments

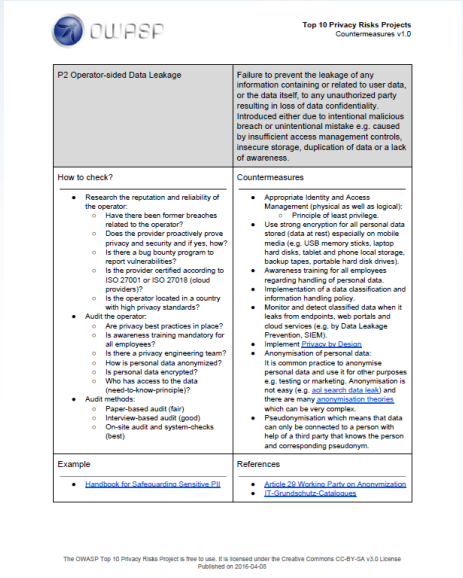
- German translation available
- Recommended by DPAs (Bavarian, Hessian?)
- Countermeasures document published in April 2016
  - Contains sections on how to check risks and improve



OWASP

Top 10 Privacy Risks Projects
Countermeasures v1.0

P1 Web Application Vulnerabilities	Vulnerability is a key problem in any system that guards or operates on sensitive user data. Failure to suitably design and implement an application, detect a problem or promptly apply a fix (patch) is likely to result in a privacy breach. This risk also encompasses the OWASP Top 10 List of web application vulnerabilities and the risks resulting from them.
How to check? • Are regular penetration tests performed with a focus on privacy? • Are developers trained regarding web application security? • Are secure coding guidelines applied? • Is any of the used software out of date (server, database, frameworks, other infrastructure components)?	Countermeasures • Perform regular penetration tests by independent security experts. • Track remediation of findings. • Train application developers and architects in secure development. • Apply procedures for secure development (e.g. Security Development Lifecycle - SDL). • Install updates, patches and hotfixes on a regular basis.
Example • Injection Flaws allow attackers among others to copy or manipulate data by attacks like SQL Injection. • Sensitive Data Exposure allows attackers gather sensitive information e.g. due to missing encryption with a man-in-the-middle attack. • Use of Insecure Direct Object References allows attackers to guess and access sensitive information, especially if access control is missing. • Usage of Components with Known Vulnerabilities, e.g. unpatched software flaws, and Security Misconfigurations, e.g. unhardened application platform. • In general it is possible for attackers to gain access to, manipulate or delete personal data that the application is processing by abusing rights, entering malicious code or eavesdropping on communications.	References • OWASP Top 10 Project • OWASP ASVS • Open SAMM • OWASP Proactive Controls • Security Development Lifecycle (SDL) • OWASP Secure Application Design Project • Lists of known vulnerabilities can be found at CVE and NVD • ISMS of the German Federal Office for Information Security (BSI)



OWASP

Top 10 Privacy Risks Projects
Countermeasures v1.0

P2 Operator-sided Data Leakage	Failure to prevent the leakage of any information containing or related to user data, or the data itself, to any unauthorized party resulting in loss of data confidentiality. Introduced either due to intentional malicious breach or unintentional mistake e.g. caused by insufficient access management controls, insecure storage, duplication of data or a lack of awareness.

How to check?
• Research the reputation and reliability of the operator:
  ◦ Have there been former breaches related to the operator?
  ◦ Does the provider proactively prove privacy and security and if yes, how?
  ◦ Is there a bug bounty program to report vulnerabilities?
  ◦ Is the provider certified according to ISO 27001 or ISO 27018 (cloud providers)?
  ◦ Is the operator located in a country with high privacy standards?
• Audit the operator:
  ◦ Are privacy best practices in place?
  ◦ Is awareness training mandatory for all employees?
  ◦ Is there a privacy engineering team?
  ◦ How is personal data anonymized?
  ◦ Is personal data encrypted?
  ◦ Who has access to the data (need-to-know-principle)?
• Audit methods:
  ◦ Paper-based audit (fair)
  ◦ Interview-based audit (good)
  ◦ On-site audit and system-checks (best)
	Countermeasures
	• Appropriate Identity and Access Management (physical as well as logical):
	  ◦ Principle of least privilege.
	• Use strong encryption for all personal data stored (data at rest) especially on mobile media (e.g. USB memory sticks, laptop hard disks, tablet and phone local storage, backup tapes, portable hard disk drives).
	• Awareness training for all employees regarding handling of personal data.
	• Implementation of a data classification and information handling policy.
	• Monitor and detect classified data when it leaks from endpoints, web portals and cloud services (e.g. by Data Leakage Prevention, SIEM).
	• Implement Privacy by Design.
	• Anonymisation of personal data:
	It is common practice to anonymise personal data and use it for other purposes e.g. testing or marketing. Anonymisation is not easy (e.g. aol search data leak) and there are many anonymisation theories which can be very complex.
	• Pseudonymisation which means that data can only be connected to a person with help of a third party that knows the person and corresponding pseudonym.

Example • Handbook for Safeguarding Sensitive PII	References • Article 29 Working Party on Anonymization • IT-Grundschutz-Catalogues

OWASP
Open Web Application Security Project

P1: Web Application Vulnerabilities

How to check?

- Are regular penetration tests performed focusing on privacy?
- Are developers trained regarding web application security?
- Are secure coding guidelines applied?
- Is any of the used software out of date (server, DB, libraries)?

How to improve?

- Apply procedures like the Security Development Lifecycle (SDL)
- Perform regular penetration tests by independent experts
- Install updates, patches and hotfixes on a regular basis

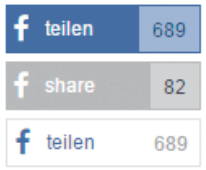
P7: Sharing of Data with 3rd Party

How to check?

- Are third party solutions in use (plugins, buttons, maps, videos, advertising, etc.), which ones and what personal data is transferred?
- Is third party tracking disclosed (which third parties and what data)?
- Are third parties rated and checked regarding privacy?
- Is privacy and handling of personal data part of the contract and if yes, what restrictions are in place?

How to improve?

- Use third party content only where required, not by default
- Develop a Third Party Monitoring Strategy
- Use privacy friendly solutions like
  - Social networks buttons that only send data on click (heise Shariff)
  - Youtube enhanced privacy mode
  - …



heise Shariff: https://github.com/heiseonline/shariff

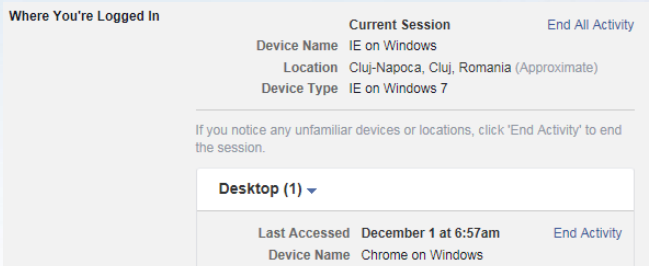
P9: Missing or Insufficient Session Expiration

How to check?

- Is there an automatic session timeout < 1 week (for critical applications < 1 day).
- Is the logout button easy to find and promoted?

How to improve?

- Configure to automatically logout after X hours / days or user-defined
- Obvious logout button
- Educate users



Picture sources: facebook.com, web.de

Further information

- OWASP Top 10 Privacy Risks Project: https://www.owasp.org/index.php/OWASP_Top_10_Privacy_Risks_Project
- Project sponsor: http://www.msg-systems.com
- My personal blog: http://securitybydesign.de/

Sponsored by

OWASP Open Web Application Security Project