DevStaff Meetup #7

InfoSec:
Developing with Security in Mind
a.k.a AppSec

Lefteris Stavrakakis

- ➔ IT Support - SysAdmin (in the before time)
- ➔ Also dealt with critical infrastructure - airports
- ➔ ISO:27001:2013 Lead Auditor
- ➔ Implementing ISMS

Lefteris@cnet.gr

@LeftyDidi

Information Security, not.

Information Security

Confidentiality

The protection of information from unauthorised disclosure

Integrity

The accuracy and completeness of information in accordance with business values and regulations

Availability

The ability to access information and resources required by the business process

Moar Buzzwords!

Information Security deals with information, regardless of its format. It includes:

- ➔ Paper documents
- ➔ Digital and intellectual property
- ➔ Verbal or visual communications

Cyber Security is concerned with protecting digital assets. Includes:

- ➔ Networks
- ➔ Hardware
- ➔ Software
- ➔ Information that is processed, stored or transported by “internetworked” Information Systems

Breaches

Who?	How big?	When?	What?
YouPorn	1.327.567	Feb 2012	Email, passwords
Domino's	648.231	June 2014	Email addresses, home addresses, names, passwords, phone numbers
vBulletin	518.966	Nov 2015	Dates of birth, email addresses, homepage URLs, instant messenger identities, IP addresses, passwords, security questions and answers, spoken languages, website activity
Vodafone	56.021	Nov 2013	Credit cards, email addresses, government issued IDs, home addresses, IP addresses, names, passwords, phone numbers, purchases, SMS messages, usernames
Sony	37.103	*2011*	Dates of birth, email addresses, genders, home addresses, names, passwords, phone numbers, usernames

Just a couple more..

Ashley Madison - July 2015

The attackers threatened Ashley Madison with the full disclosure of the breach unless the service was shut down. One month later, the database was dumped, **30.8 million accounts**.

Compromised data: dates of birth, email addresses, ethnicities, genders, home addresses, names, passwords, payment histories, phone numbers, security questions and answers, sexual preferences, usernames, website activity

Adobe - October 2013

The big one.

152.4 million accounts were breached. The password cryptography was poorly done and many were quickly resolved back to plain text.

Compromised data: email addresses, password hints, passwords, usernames

Bummer, why should I care?

Because, European Data Protection Directive (Directive 95/46/EC)

Replaced by, **General Data Protection Regulation (GDPR)**

Also, due to the Network and Information Security (NIS) Directive

...and a bunch of other stuff that's on the way.

GDPR

➔ Definitions of Personal Data

"'personal data' means any information relating to an identified or identifiable natural person 'data subject'; an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that person"

➔ Personal Data Breaches

"breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed"

➔ Sanctions

- ◆ effective - proportionate - dissuasive

"up to 20.000.000 EUR, or in case of an undertaking, up to 4% of the total worldwide annual turnover of the preceding financial year, whichever is **higher**"

Other areas of interest:

- ➔ Marketing consent
- ➔ Notification and Legal Processing
- ➔ Legal rights of Data Subjects
- ➔ Data Protection Officer (DPO)

GDPR

1. Don’t panic
2. Assess true impact
3. Prioritise accountability
4. Think strategically about dataflows
5. See it as an opportunity

Ok, now what?

The **Open Web Application Security Project (OWASP)** is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted.

OWASP

- ➔ Application security tools and standards
- ➔ Complete books on application security testing, secure code development, and secure code review
- ➔ Standard security controls and libraries
- ➔ Local chapters worldwide
- ➔ Cutting edge research
- ➔ Extensive conferences worldwide

Here be dragons ‘n stuff.

Please welcome DaKnOb :)

Uhm, I do that stuff already. Really!

The **Application Security Verification Standard (ASVS)** is a list of application security requirements or tests that can be used to define what a secure application is, it has two main goals:

1. To help organizations develop and maintain secure applications.
2. To allow security service, security tools vendors, and consumers to align their requirements and offerings

ASVS

ASVS defines three security verification levels

- ➔ **ASVS Level 1** is meant for all software.
- ➔ **ASVS Level 2** is for applications that contain sensitive data, which requires protection.
- ➔ **ASVS Level 3** is for the most critical applications - applications that perform high value transactions, contain sensitive medical data, or any application that requires the highest level of trust.

ASVS

Level 1: Opportunistic

- ➔ Easy to discover
- ➔ Included in the OWASP Top 10
- ➔ Quick analysis of many applications
- ➔ Can be ensured automatically
- ➔ Do not need access to source code
- ➔ Considered minimum required

ASVS

Level 2: Standard

- ➔ Defends against most risks (associated with software)
- ➔ Ensures that security controls are in place, effective, and used
- ➔ Significant business-to-business transactions
- ➔ Skilled and motivated attackers

ASVS

Level 3: Advanced

- ➔ Military, health and safety, critical infrastructure, etc.
- ➔ Applications that perform critical functions, where failure is a “no-no”
- ➔ Wire transfers (large sums of money)
- ➔ Possibility of threatening safety of life
- ➔ Medical equipment
- ➔ Large volume of sensitive information
- ➔ Covers PCI DSS, HIPAA, SOX Act, ISO27k, and pretty much everything.

Detailed Verification Requirements

V1. Architecture, design and threat modelling

V2. Authentication

V3. Session management

V4. Access control

V5. Malicious input handling

V7. Cryptography at rest

V8. Error handling and logging

V9. Data protection

V10. Communications

V11. HTTP security configuration

V13. Malicious controls

V15. Business logic

V16. File and resources

V17. Mobile

V18. Web services

V19. Configuration

V1: Architecture, design and threat modelling

#	Description	1	2	3	Since
1.1	Verify that all application components are identified and are known to be needed.	✓	✓	✓	1.0
1.2	Verify that all components, such as libraries, modules, and external systems, that are not part of the application but that the application relies on to operate are identified.		✓	✓	1.0
1.3	Verify that a high-level architecture for the application has been defined.		✓	✓	1.0
1.4	Verify that all application components are defined in terms of the business functions and/or security functions they provide.			✓	1.0
1.5	Verify that all components that are not part of the application but that the application relies on to operate are defined in terms of the functions, and/or security functions, they provide.			✓	1.0
1.6	Verify that a threat model for the target application has been produced and covers off risks associated with Spoofing, Tampering, Repudiation, Information Disclosure, and Elevation of privilege (STRIDE).			✓	1.0
1.7	Verify all security controls (including libraries that call external security services) have a centralized implementation.			✓	1.0
1.8	Verify that components are segregated from each other via a defined security control, such as network segmentation, firewall rules, or cloud based security groups.		✓	✓	3.0

V2: Authentication Verification Requirements

#	Description	1	2	3	Since
2.1	Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓	1.0
2.2	Verify that all password fields do not echo the user's password when it is entered.	✓	✓	✓	1.0
2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓	1.0
2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓	1.0
2.7	Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent long passphrases/highly complex passwords being entered.	✓	✓	✓	3.0
2.8	Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.	✓	✓	✓	2.0
2.9	Verify that the changing password functionality includes the old password, the new password, and a password confirmation.	✓	✓	✓	1.0
2.12	Verify that all suspicious authentication decisions are logged. This should include requests with relevant metadata needed for security investigations.		✓	✓	2.0

V8: Error handling and logging verification requirements

#	Description	1	2	3	Since
8.1	Verify that the application does not output error messages or stack traces containing sensitive data that could assist an attacker, including session id, software/framework versions and personal information	✓	✓	✓	1.0
8.2	Verify that error handling logic in security controls denies access by default.		✓	✓	1.0
8.3	Verify security logging controls provide the ability to log success and particularly failure events that are identified as security-relevant.		✓	✓	1.0
8.4	Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens.		✓	✓	1.0
8.5	Verify that all events that include untrusted data will not execute as code in the intended log viewing software.			✓	1.0
8.6	Verify that security logs are protected from unauthorized access and modification.		✓	✓	1.0
8.7	Verify that the application does not log sensitive data as defined under local privacy laws or regulations, organizational sensitive data as defined by a risk assessment, or sensitive authentication data that could assist an attacker, including user's session identifiers, passwords, hashes, or API tokens.		✓	✓	3.0

Getting started

- ➔ Agree on ASVS Level
- ➔ Perform initial review
- ➔ Develop verification plan and project schedule
- ➔ Stick to them - “or adapt and overcome”
- ➔ Present findings
- ➔ Fix ‘em
- ➔ Re-verify after fixes.
- ➔ Ideally, develop a strategy to add verifications into the SDLC

References, additional info/resources.

Just google (or duckduckgo):

- ➔ OWASP Top 10 2013
- ➔ OWASP ASVS
- ➔ GDPR

Tweets:

@manicode - ASVS co-author, AppSec, etc.

@kpapapan - OWASP Athens chapter leader

Thank you!

Antonis | Lefteris