2017: OWASP Top 10

Dirk Wetter

@drwetter

License: http://creativecommons.org/licenses/by-nc-sa/4.0/

Was ist OWASP?

- **OWASP**
  - Foundation (501c3), Steuervorteile
    - *worldwide not-for-profit charitable organization focused on improving the security of software*
  - Vgl. Suppenküche / Gemeinnützigkeit
    - Selbstlos: deutsches Steuergesetz
  - Voll Open Source
    - Kein Open Core / Dual License
  - Rolle Vendors / Hersteller

OWASP Open Web Application Security Project

0. Einleitung

Worum geht‘s?

- **OWASP Top 10 2017**
  - Was ist das (und was nicht?)
  - Geschichte, Hintergründe
  - Was gibt’s Neues?
  - Kritik +/-

Motivation

- **Monty Python / Leben des Brian**
  - Der Messias :-)

OWASP Open Web Application Security Project

1. Definition

Das wichtigste vorweg!

- **Definition ?**

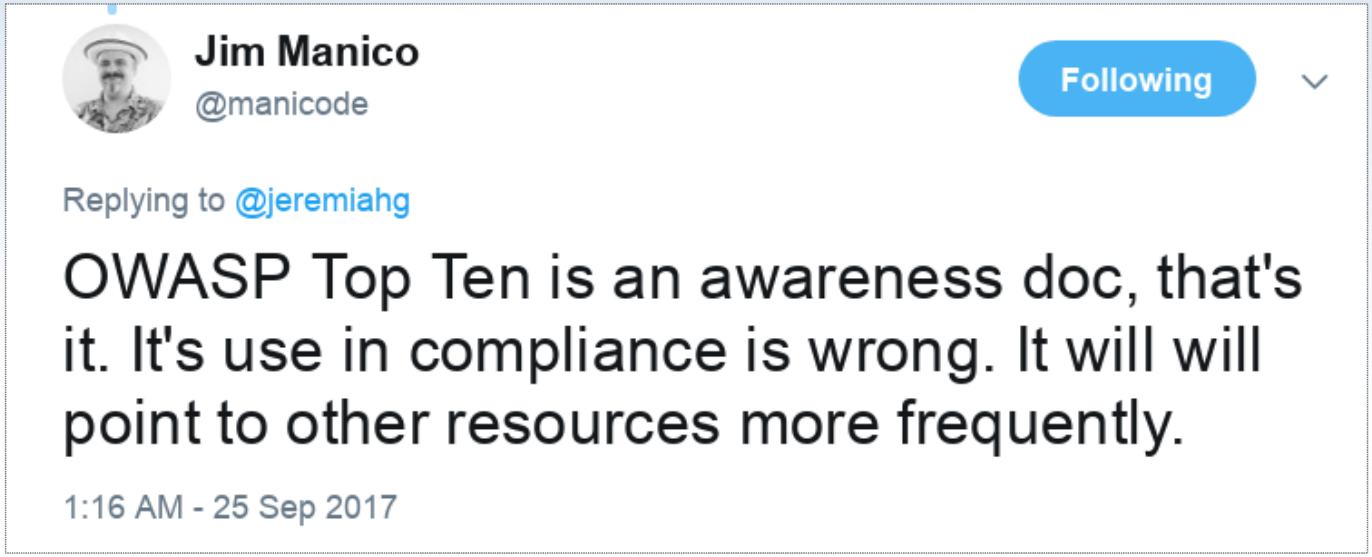
OWASP Open Web Application Security Project

1. Definition

Das wichtigste vorweg!

- Definition!

- **Awareness-Dokument !**



1. Definition

- **Keine** Compliance
- **Kein** Risiko-Management
- Beschreibt **keinen** SDLC oder BS-irgendwas
- **Keine** Pentest-Schablone

→ **Einstieg** in Application Security!

- Verkaufsargument?

OWASP Open Web Application Security Project

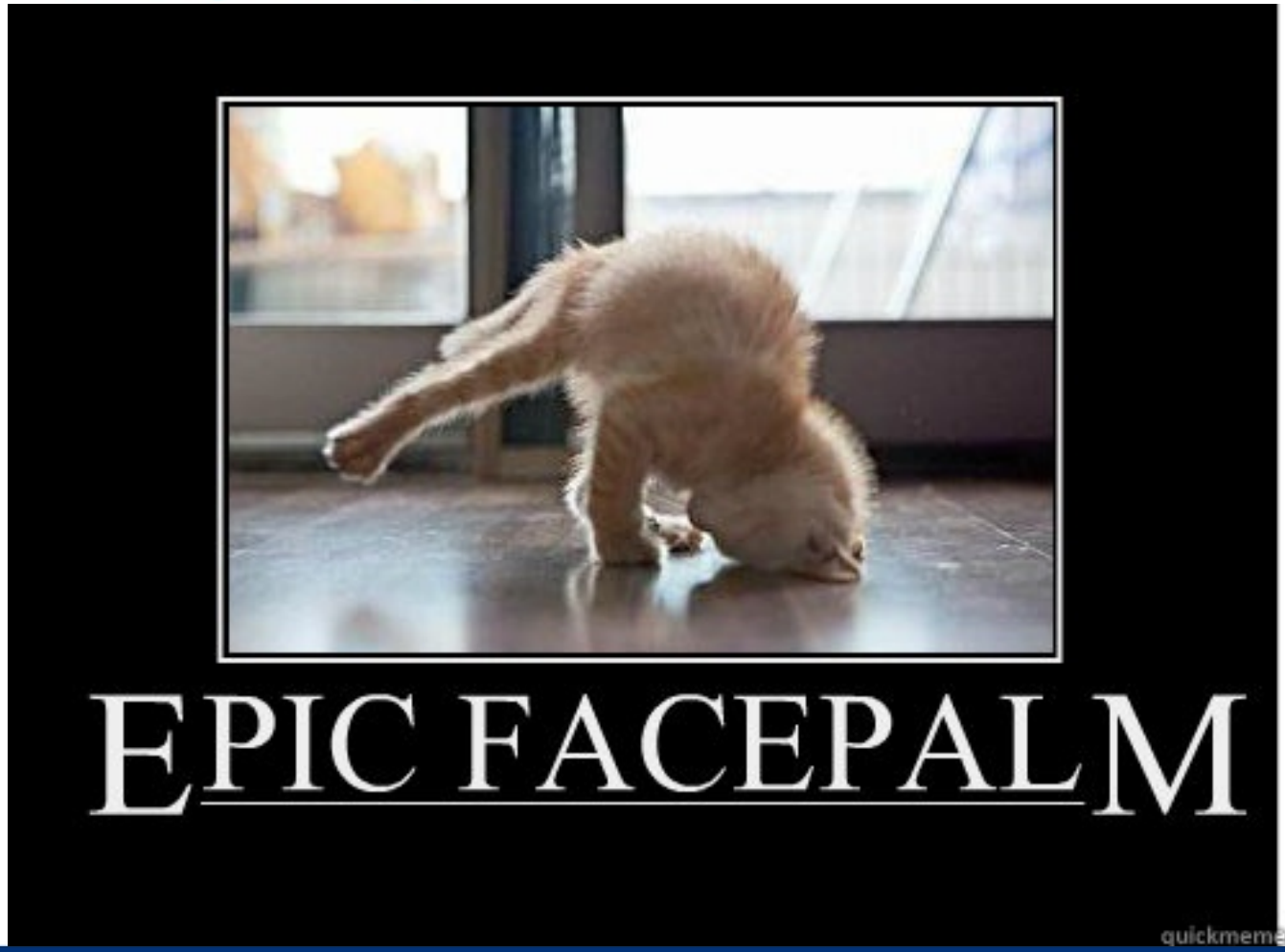
1. Definition

BS Bingo!!!

Tom Brennan
@brennantom

Following

The NEW @owasp Top 10 2017 needs to be considered as part of your SDLC #AllDayDevOps



OWASP Open Web Application Security Project

1. Definition

A-W-A-R-E-N-E-S-S – DOKUMENT !!!

- **Monty Python / Leben des Brian:**
  - Bitte A-W-A-R-E-N-E-S-S an die Wand schreiben (wie "Romani ite domum")

1. Definition

Ggw. Stellungnahme

At OWASP, we want organizations and developers to adopt actual standards, [..], but we understand that the OWASP Top 10 is the most downloaded, most referenced, and ***for better or worse, most improperly-used and abused OWASP document****.*

1. Definition

42 ?!

- **Warum → 10 ← ??**
  - SANS / CWE: Top 25
  - WASC Threat Classification (**49**)

OWASP Open Web Application Security Project

2. Geschichte

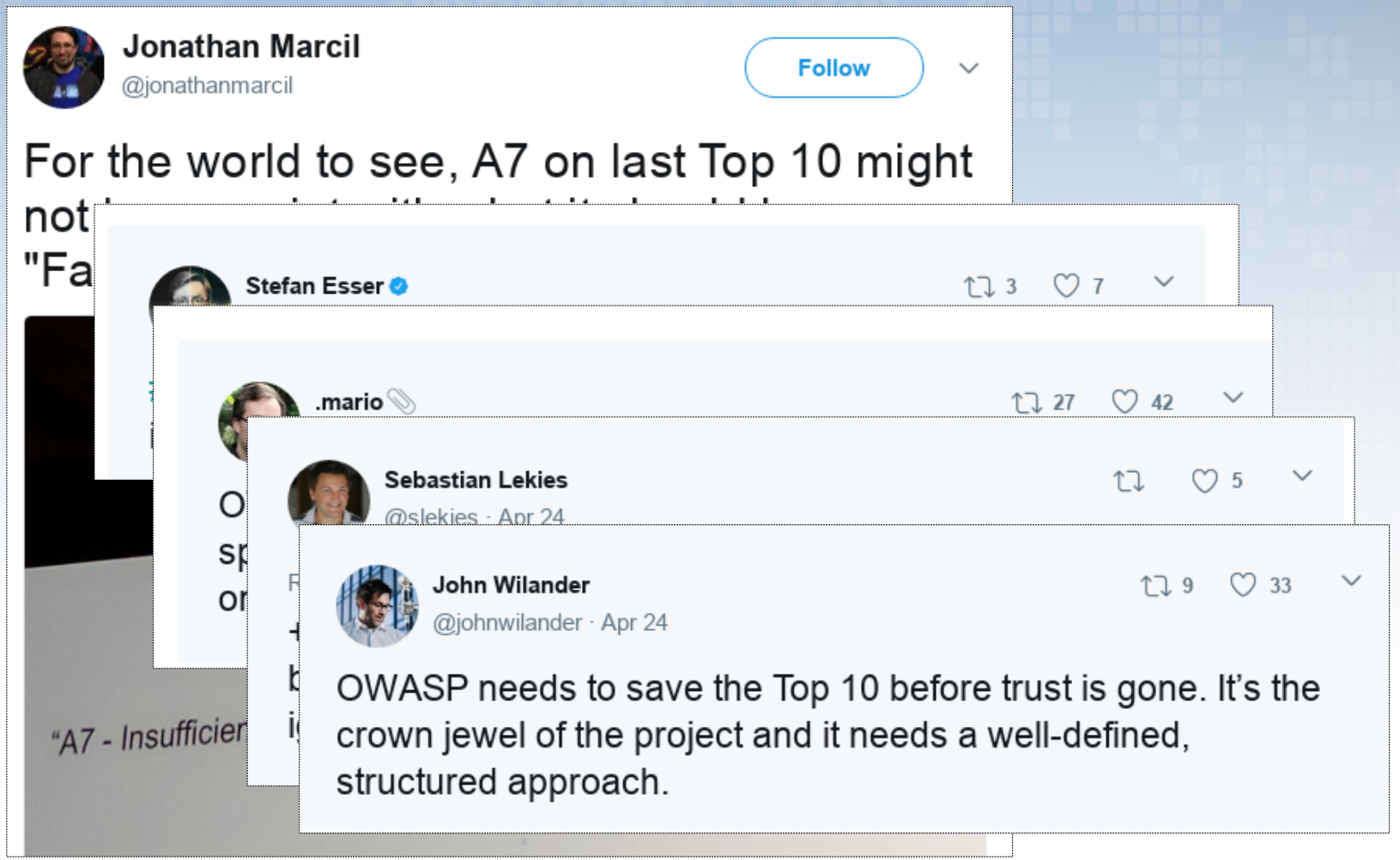
- **2017: Nr #6**
  - 2013 ← 2010 ← 2007 ← 2004 ← 2003
  - Fast hinter verschlossenen Türen von einer Firma
  - Transparenz
    - Bisher leider eher: WASP Top10 (ohne O)

2. Geschichte

- **2017: alter A7**
  - Hauptautor Top 10 in Firmenblog:

> The Two New Vulnerabilities: Insufficient Attack Protection & Underprotected APIs
>
> **A7: Insufficient Attack Protection.** This new requirement means that applications need to detect, prevent, and respond to both manual and automated attacks. No longer will attackers be prompted with "Invalid input, please try again." Instead, anyone attempting attacks will have their attempts blocked and their account flagged. Web application firewalls have been ineffective at blocking application attacks because they have no context for what they are protecting. Contrast Protect effectively blocks attacks by injecting the protection directly into the application where it can take advantage of the full application context.

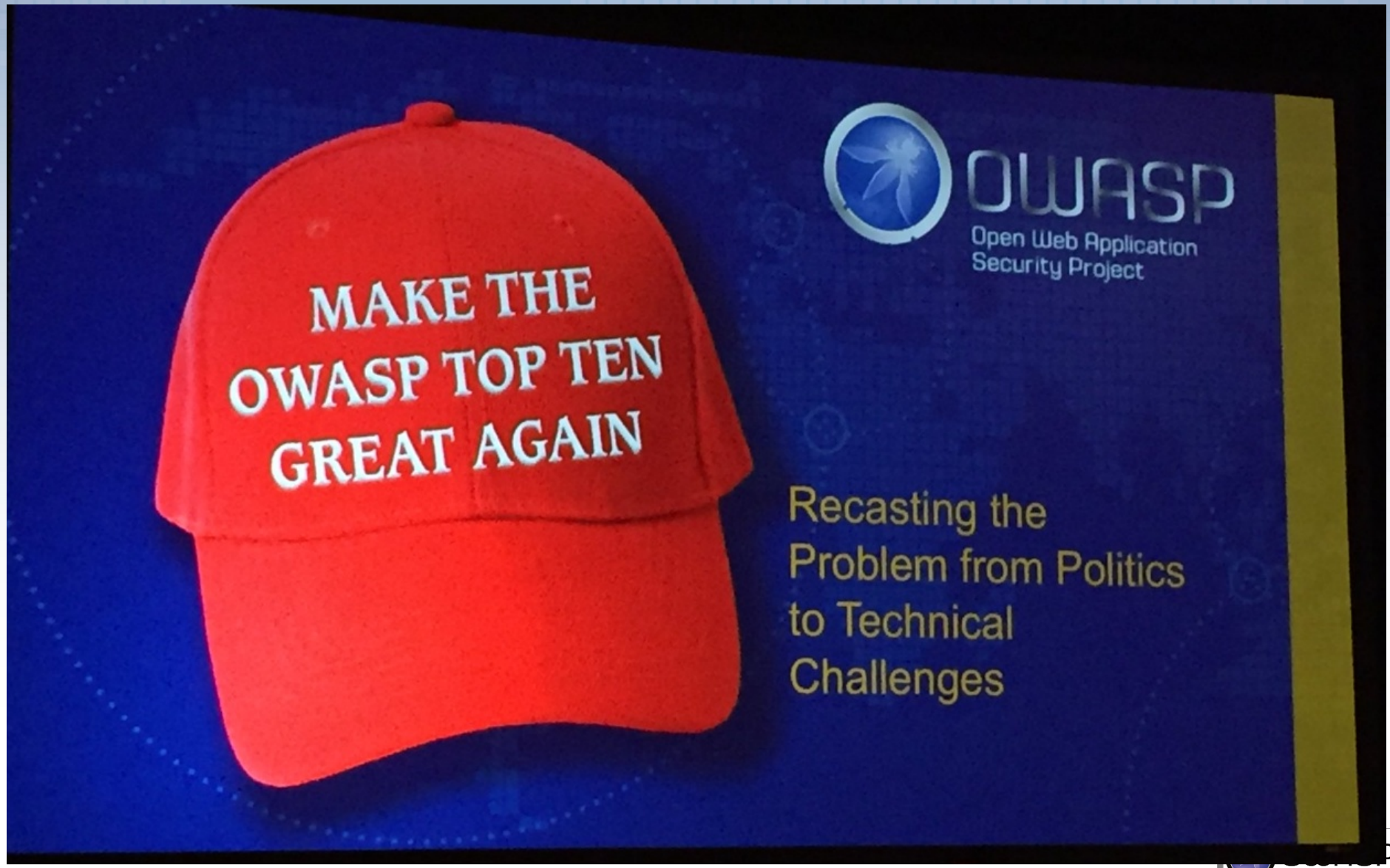
2. Geschichte



2. Geschichte

2. Geschichte

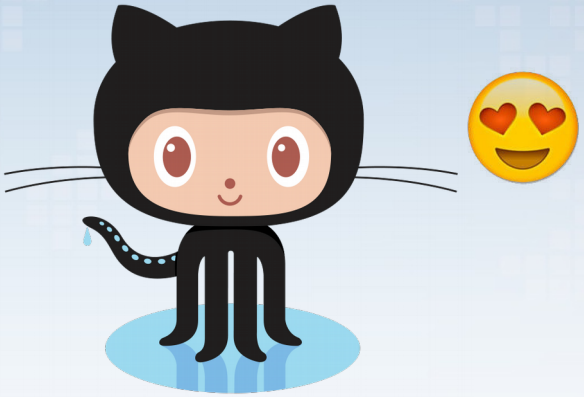
hold on while rebooting



2. Geschichte

hold on while rebooting

- **OWASP Top 10 2017**
  - Neue „Leads“
    - Community, github
    - RC2 20.10.
    - GA 21.11



  - Inhalt – wie entstehen die TT?
    - Data Call (80%)
    - „Forward Looking“: 20% / Community

3. 2017er Top 10

Wie sieht das aus?

OWASP Top 10 - 2017

The Ten Most Critical Web Application Security Risks

OWASP Open Web Application Security Project

3. 2017er Top 10

Final versions

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication (and Session Management)
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.]

OWASP Open Web Application Security Project

3. 2017er Top 10

A1 Injection

Application Security Risk	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts
	Easy: 3	Widespread: 3	Easy: 3	Severe: 3
	Average: 2	Common: 2	Average: 2	Moderate: 2
	Difficult: 1	Uncommon: 1	Difficult: 1	Minor: 1

Am I Vulnerable To

How Do I Prevent

Example Attack Scenarios

References

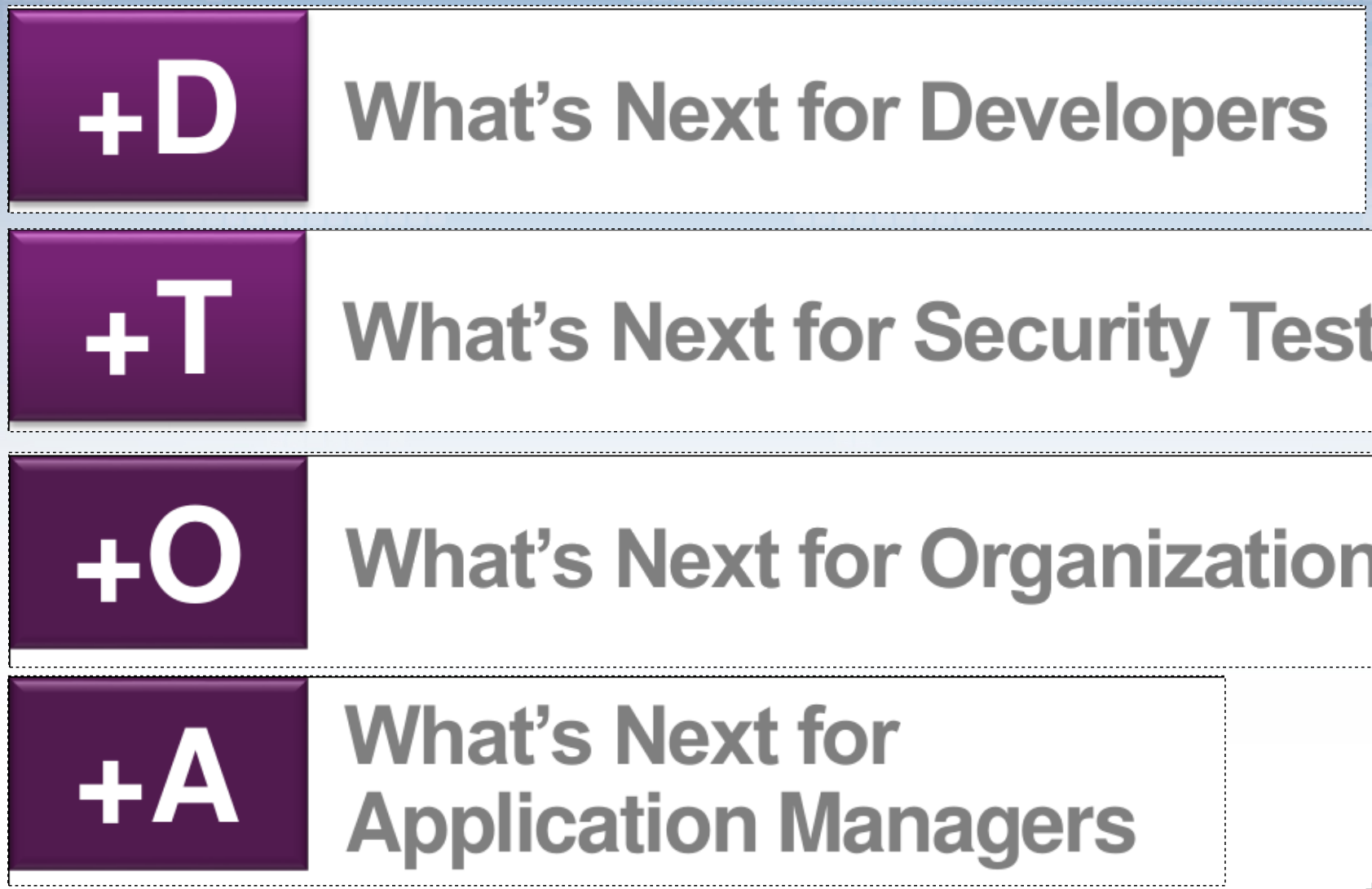
OWASP

External

OWASP Open Web Application Security Project

3. 2017er Top 10

Lesenswertes



OWASP Open Web Application Security Project

3. 2017er Top 10

- **Neu: A8 Insecure Deserialization**
- Remember Apache Commons Vuln.?
  - Nicht darauf beschränkt, auch nicht auf Java
  - Objekte vom serialisierten Format zurück
    - Untrusted
- Häufig RCE

4. Kritik

- **Datenerhebung/-interpretation**
  - Von Data Calls
  - Statistische Grundlage nur teilweise klar
  - Keine Wissenschaft!
    - Interpretation anders :-(

A4: XXE

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE test [
    <!ENTITY xxeattack SYSTEM "<ANY server or remote ressource>">
]>
<xxx>&xxeattack;</xxx>
```

file:///etc/passwd
http://evilstuff.com/evil.inc

libxml2	PHP	Java	.NET
file	file	http	file
http	http	https	http
ftp	ftp	ftp	https
	php	file	ftp
	compress.zlib	jar	
	compress.bzip2	netdoc	
	data	mailto	
	glob	gopher *	
	phar		
	express		

4. Kritik

- **XXE?**
  - XML: Dinosaurier!
  - Praxisrelevanz Daten zweifelhaft: SAST
    - HTML/JSON/XML/… Daten vs. Realität
  - Wer schreibt sich einen XML-Parser selber?
    - → A9: known vulnerable components
    - → A6: security misconfiguration

A10: Insufficient Logging & Monitoring

Is the Application Vulnerable?

Insufficient logging, detection, monitoring and active response occurs any time:

- Auditable events, such as logins, failed logins, and high-value transactions are not logged.
- Warnings and errors generate no, inadequate, or unclear log messages.
- Logs of applications and APIs are not monitored for suspicious activity.
- Logs are only stored locally.
- Appropriate alerting thresholds + response escalation processes are not in place or effective.
- Penetration testing and scans by DAST tools (such as OWASP ZAP) do not trigger alerts.
- The application is unable to detect, escalate, or alert for active attacks [..] near real time.

OWASP Open Web Application Security Project

4. Kritik

- **A10: Insufficient Logging & Monitoring**
- **Das alte A7?**
  - Hmm, wird aber sicherlich Vendors geben

4. Kritik

- **A10: Insufficient Logging & Monitoring**
- **Das alte A7?**
  - Hmm, wird aber sicherlich Vendors geben
  - Snake Oil (#fefe)

Dafür (A7) ist da "du sollst Schlangenöl zum Monitoring + Alerting kaufen" drin. Tja, Owasp, ein Wort mit X. Das war wohl nix.

4. Kritik

- **A10: Insufficient Logging & Monitoring**
  - Logging immens wichtig, aber **Risiko**??
    - Erklärungen IMO schwach
    - Einbrecher vs. Video-Kamera
    - Sauberes Logging: schwierig!
    - Korrelation noch mehr (SIEM)
    - Unterscheidung Versuch / Erfolg
      - Diskussion IDS vs. IPS

4. Kritik

- **A10: Insufficient Logging & Monitoring**
  - Logging immens wichtig, aber **Risiko**??
- Meine 

  - Risikoargumentation dürftig. Nicht als Risiko verkaufen, sonst raus.

Ich finde es halt anstößig, erst die Formulierung von wichtigste " Angriffe" auf "wichtigste Risiken" zu ändern, um dann in der nächsten Runde "Reaktives Security-Produkt $XY nicht installiert" als Risiko hinzu-Schreiben. Das geht aus meiner Sicht gar nicht (#fefe)

4. Kritik

- **A2: Session Management?**
  - 

    Mal raus / mal rein
  - Ggw: jain ;-)
    - Titel “Broken Authentication”
    - Eingestreut: Session Management

Is the Application Vulnerable?

- Exposes Session IDs in the URL (e.g., URL rewriting).
- Does not rotate Session IDs after successful login.
- Does not properly invalidate Session IDs. User sessions or authentication tokens (particularly single sign-on (SSO) tokens) aren’t properly invalidated during logout or a period of inactivity.

JWT und Freunde

How to Prevent

- Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login. Session IDs should not be in the URL, be securely stored and invalidated after logout, idle, and absolute timeouts.

4. Kritik

- **A2**
  - Mehr Gewicht auf Broken Authentication
  - Credential Stuffing
    - aus breaches
    - → MFA, PW checks aus breaches
  - Neue NIST PW-Regeln: Komplexität+ Expiration ~out
    - *Password length has been found to be a primary factor in characterizing password strength*
    - *at least 8 [..], SHOULD permit [..] at least 64 char[.]s*
    - *MAY issue authenticators that expire.*
  - Eigentlich fehlt nur im Titel “Session Management”

3. 2017er Top 10

Status 2017 RC2

- **Mehr?**
  - Details A1-A10 → Klimperkisteninhibitor
  - Empfehlung Top 10
    - Alles 1x Durchlesen
    - Nicht nur A1-10, D, T, O, A
    -  , bald wieder: 

- **Jeder nur ein Kreuz...**

- **Danke!**

  mail bei drwetter punkt eu

  dirk aet owasp org

  @drwetter