OWASP Stammtisch #37
Intro
Intro

• My name is ...

• I work as ...

• I’m here because ...
Agenda

1. Talk – Johannes: What is OWASP? (why am I here?)

2. Next meetup
What is OWASP?

Open Web Application Security Project

- worldwide free and open community focused on improving the security of application software
- Promotes secure software development
- Oriented to the delivery of web oriented services
- An open forum for discussion
- A free resource for any development team
What is OWASP?

• Non-profit (501c3), volunteer driven organization
  • All members are volunteers (save 4 employees)
  • All work is donated by volunteers and sponsors

• Provide free resources to the community
  • Publications, Articles, Standards
  • Testing and Training Software
  • Local Chapters & Mailing Lists

• Supported through sponsorships
  • Corporate support through financial or project sponsorship
  • Personal sponsorships from members
OWASP Organization

• Global Board
• Global Committees
  – Education
  – Chapters
  – Conferences
  – Industry
  – Projects & Tools
  – Membership
• Employees
• Volunteers
## OWASP membership

<table>
<thead>
<tr>
<th>Membership category</th>
<th>Annual membership fee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Supporters</td>
<td>$50</td>
</tr>
<tr>
<td>Organization Supporters</td>
<td>$5,000</td>
</tr>
<tr>
<td>Accredited University Supporters</td>
<td>FREE (in exchange of meeting space at least 2x per year)</td>
</tr>
<tr>
<td>Lifetime Membership</td>
<td>$500</td>
</tr>
</tbody>
</table>

- Funds OWASP Speakers via OWASP On the Move
- Funds Season of Code projects
- Helps Support Local Chapters
Talk

I'm Lost...

How Does It Work?
Talk

Write Secure Code → Audit Code, Result → Control Risk
Talk

Write Secure Code → Audit Code, Result → Control Risk

- Software Assurance Maturity Model (SAMM)
- Mobile Application Security Verification Standard (MASVS)
- Cheat Sheet Series
- ...

OWASP Frankfurt Stammtisch #37
Talk

Software Development

SAMM Overview

Business Functions
- Governance
- Construction
- Verification
- Operations

Security Practices
- Strategy & Metrics
- Education & Guidance
- Security Requirements
- Design Review
- Security Testing
- Environment Hardening
- Policy & Compliance
- Threat Assessment
- Secure Architecture
- Implementation Review
- Issue Management
- Operational Enablement

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Talk

Write Secure Code → Audit Code, Result → Control Risk

- Software Assurance Maturity Model (SAMM)
- Mobile Application Security Verification Standard (MASVS)
- Cheat Sheet Series
- ...

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
### Description

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>L1</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>System credential storage facilities are used appropriately to store sensitive data, such as user credentials or cryptographic keys.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2.2</td>
<td>No sensitive data is written to application logs.</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

No sensitive data is shared with third parties unless it is necessary.
Talk

Write Secure Code ➔ Audit Code, Result ➔ Control Risk

• Software Assurance Maturity Model (SAMM)
• Mobile Application Security Verification Standard (MASVS)
• Cheat Sheet Series
• ...

OWASP Frankfurt Stammtisch #37
## OWASP Cheat Sheets

<table>
<thead>
<tr>
<th>Category</th>
<th>Cheat Sheets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile</td>
<td>Android Testing, IOS Developer, Mobile Jailbreaking</td>
</tr>
<tr>
<td>OpSec/Defender</td>
<td>Virtual Patching</td>
</tr>
</tbody>
</table>

All Pages In This Category
Safe Java Stored Procedure Example

The following code example uses a CallableStatement, Java's implementation of the stored procedure interface, to execute the same database query. The "sp_getAccountBalance" stored procedure would have to be predefined in the database and implement the same functionality as the query defined above.

```java
String custname = request.getParameter("customerName"); // This should REALLY be validated
try {
    CallableStatement cs = connection.prepareCall("{call sp_getAccountBalance(?)}");
    cs.setString(1, custname);
    ResultSet results = cs.executeQuery();
    // ... result set handling
} catch (SQLException se) {
    // ... logging and error handling
}
```
Talk

Write Secure Code ➔ Audit Code, Result ➔ Control Risk

- OWASP Testing Guide
- «OWASP Top10» Web, Mobile, IoT...
- OWASP Zed Attack Proxy
- ...

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Testing for Error Handling
Analysis of Error Codes (OTG-ERR-001)
Analysis of Stack Traces (OTG-ERR-002)

Testing for weak Cryptography
Testing for Weak SSL/TLS Ciphers, Insufficient Transp
Testing for Padding Oracle (OTG-CRYPT-002)
Testing for Sensitive information sent via unencrypted

Authorization Testing
Testing Directory traversal/file include (OTG-AUTHZ-001)
Testing for bypassing authorization schema (OTG-AUTHZ-002)
Testing for Privilege Escalation (OTG-AUTHZ-003)
Testing for Insecure Direct Object References (OTG-AUTH

Session Management Testing
Testing for Bypassing Session Management Schema (OTG-
Testing for Cookies attributes (OTG-SESS-002)

Identity Management Testing
Test Role Definitions (OTG-IDENT-001)
Test User Registration Process (OTG-IDENT-002)
Test Account Provisioning Process (OTG-IDENT-003)
Testing for Account Enumeration and Guessable User Account (OTG-IDENT-004)
Testing for Weak or unenforced username policy (OTG-IDENT-005)

Authentication Testing
Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)
Testing for default credentials (OTG-AUTHN-002)
Testing for Weak lock out mechanism (OTG-AUTHN-003)
Testing for bypassing authentication schema (OTG-AUTHN-004)
Authorization Testing
Testing Directory traversal/file include (OTG-AUTHZ-001)

Testing Techniques
The next stage of testing is analyzing the input validation functions present in the web application. Using the previous example, the dynamic page called getUserProfile.jsp loads static information from a file and shows the content to users. An attacker could insert the malicious string “../../../../etc/passwd” to include the password hash file of a Linux/UNIX system. Obviously, this kind of attack is possible only if the validation checkpoint fails; according to the file system privileges, the web application itself must be able to read the file.

To successfully test for this flaw, the tester needs to have knowledge of the system being tested and the location of the files being requested. There is no point requesting /etc/passwd from an IIS web server.

http://example.com/getUserProfile.jsp?item=../../../../etc/passwd
Talk

FULL OF WIN!

memecenter.com
Talk

Write Secure Code → Audit Code, Result → Control Risk

- OWASP Testing Guide
- "OWASP Top10" Web, Mobile, IoT...
- OWASP Zed Attack Proxy
- ...

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/asset/112705554/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
For 2017, the OWASP Top 10 Most Critical Web Application Security Risks (in the Release Candidate) are:

- A1 Injection
- A2 Broken Authentication and Session Management
- A3 Cross-Site Scripting (XSS)
- A4 Broken Access Control (As it was in 2004)
- A5 Security Misconfiguration
- A6 Sensitive Data Exposure
- A7 Insufficient Attack Protection (NEW)
- A8 Cross-Site Request Forgery (CSRF)
- A9 Using Components with Known Vulnerabilities
- A10 Underprotected APIs (NEW)

The OWASP Top 10 IoT Vulnerabilities from 2014 are as follows:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>Insecure Web Interface</td>
</tr>
<tr>
<td>I2</td>
<td>Insufficient Authentication/Authorization</td>
</tr>
<tr>
<td>I3</td>
<td>Insecure Network Services</td>
</tr>
<tr>
<td>I4</td>
<td>Lack of Transport Encryption/Integrity Verification</td>
</tr>
<tr>
<td>I5</td>
<td>Privacy Concerns</td>
</tr>
<tr>
<td>I6</td>
<td>Insecure Cloud Interface</td>
</tr>
<tr>
<td>I7</td>
<td>Insecure Mobile Interface</td>
</tr>
<tr>
<td>I8</td>
<td>Insufficient Security Configurability</td>
</tr>
<tr>
<td>I9</td>
<td>Insecure Software/Firmware</td>
</tr>
<tr>
<td>I10</td>
<td>Poor Physical Security</td>
</tr>
</tbody>
</table>
Talk

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
<table>
<thead>
<tr>
<th>Threat Agents</th>
<th>Attack Vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Specific</strong></td>
<td><strong>Exploitability EASY</strong></td>
</tr>
<tr>
<td>Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.</td>
<td>Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter. Almost any source of data can be an injection vector, including internal sources.</td>
</tr>
</tbody>
</table>
Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.
<table>
<thead>
<tr>
<th>Technical Impacts</th>
<th>Business Impacts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Impact</strong> SEVERE</td>
<td><strong>Application / Business Specific</strong></td>
</tr>
</tbody>
</table>

Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover.

Consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?
Injection can result in data loss, corruption, or access denied. It can also lead to lost or stolen data, and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?
Am I vulnerable to injection?  
How do I prevent injection attacks?

Example Attack Scenarios

Scenario #1: The application uses untrusted data in the construction of the following vulnerable SQL call:

```
String query = "SELECT * FROM accounts WHERE custID='' + request.getParameter("id") + "'";
```

References

OWASP

- OWASP SQL Injection Prevention Cheat Sheet
- OWASP Query Parameterization Cheat Sheet
Talk

[Diagram of Injection]

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Talk

FULL OF WIN!

memecenter.com
Talk

- Write Secure Code
- Audit Code, Result
- Control Risk

- OWASP Testing Guide
- «OWASP Top10» Web, Mobile, IoT...
- OWASP Zed Attack Proxy
- ...

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Write Secure Code, Audit Code, Result Control Risk

• OWASP Testing Guide
• «OWASP Top10» Web, Mobile, IoT …
• OWASP Zed Attack Proxy
Talk

Write Secure Code → Audit Code, Result → Control Risk

- OWASP Risk Rating Methodology
- Threat Risk Modeling
- OWASP Application Security Guide For CISOs Project
- ...

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/assets/images/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Talk
Talk

Attacker may be able to read other users' messages

- User may not have logged off on a shared computer
- Data validation may fail, allowing SQL injection
- Authorization may fail, allowing unauthorized access
- Browser cache may contain contents of message

Implement data validation
Implement authorization checks
Implement anticaching HTTP headers
If risk is high, use SSL

Write Secure Code, Audit Code, Result Control Risk
<table>
<thead>
<tr>
<th>Threat agent factors</th>
<th>Likelihood</th>
<th>Vulnerability factors</th>
<th>Intrusion detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skill level</td>
<td>4 - Advanced computer user</td>
<td>3,375</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Motive</td>
<td>1 - Low or no reward</td>
<td>3 - Difficult</td>
<td>3 - Difficult</td>
</tr>
<tr>
<td>Opportunity</td>
<td>access or resources required</td>
<td>4 - Hidden</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>5 - Partners</td>
<td>4 - Hidden</td>
<td></td>
</tr>
</tbody>
</table>

**Overall likelihood:** 3,375  **MEDIUM**

<table>
<thead>
<tr>
<th>Technical Impact</th>
<th>Overall technical impact</th>
<th>Overall business impact</th>
<th>Overall impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss of confidentiality</td>
<td>2 - Minimal non-sensitive data disclosed</td>
<td>0 -</td>
<td>2,750</td>
</tr>
<tr>
<td>Loss of integrity</td>
<td>0 -</td>
<td>0 -</td>
<td>9 - Completely anonymous</td>
</tr>
<tr>
<td>Loss of availability</td>
<td>0 -</td>
<td>0 -</td>
<td>9 - Completely anonymous</td>
</tr>
<tr>
<td>Loss of accountability</td>
<td>0 -</td>
<td>0 -</td>
<td>9 - Completely anonymous</td>
</tr>
<tr>
<td>Financial damage</td>
<td>1 - Less than the cost to fix the vulnerability</td>
<td>1 - Minimal damage</td>
<td>0 -</td>
</tr>
<tr>
<td>Reputational damage</td>
<td>1 -</td>
<td>0 -</td>
<td>1 - Minimal damage</td>
</tr>
<tr>
<td>Privacy violation</td>
<td>5 - Hundreds of people</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall technical impact:</td>
<td>2,750</td>
<td><strong>LOW</strong></td>
<td></td>
</tr>
<tr>
<td>Overall business impact:</td>
<td>1,750</td>
<td><strong>LOW</strong></td>
<td></td>
</tr>
<tr>
<td>Overall impact:</td>
<td>2,250</td>
<td><strong>LOW</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Overall Risk Severity = Likelihood x Impact**

**Likelihood and Impact Levels**

- HIGH
- MEDIUM
- LOW

- Medium
- High
- Critical

- Low
- Medium
- High

- Note
- Low
- Medium

- LOW
- MEDIUM
- HIGH

- 0 to <3
- 3 to <6
- 6 to 9

- LOW
- MEDIUM
- HIGH

---

OWASP Frankfurt Stammtisch #37

https://media.mnn.com/2016/02/bionic-arm.jpg.653x0_q80_crop-smart.jpg
Talk

FULL OF WIN!

memecenter.com
Talk

Is your app “this” secure, or “THIS” secure?

Read ASVS.

http://www.owasp.org/index.php/ASVS
Feedback
Next meetup
29.11.2017
???
Spread the word

• Mailinglisten
  – OWASP Deutschland
    • https://lists.owasp.org/pipermail/owasp-germany/
  – Stammtisch Frankfurt
    • https://lists.owasp.org/mailman/listinfo/owasp-frankfurt

• Meetup
  – Stammtisch Frankfurt
    http://www.meetup.com/de/IT-Security-Stammtisch-Frankfurt-OWASP-u-w/

• OWASP Germany
  – https://www.owasp.org/index.php/Germany
  – https://www.owasp.org/index.php/OWASP_German_Chapter_Stammtisch_Initiative
  – https://www.owasp.org/index.php/Germany/Projekte
  – https://twitter.com/#!/search/OWASP_de
Outro

Bier ist gut ... sagt der Arzt!
Danke
Quellen

• http://www.open-forum.de/events/storytelling-open-forum-rundgespraeche-procedere-FES-2008_html_m2da1a877.gif
• http://www.luxusteich.de/img/idee.jpg
• http://static.one-schweiz.ch/1335769457/roundtable_shutterstock.jpg
• http://www.couchpotatoshop.de/images/produkte/i80/801-801-801-Brillenputztuch-Wiedersehen-0.jpg
• http://www.bier-fibel.de/wp-content/uploads/2008/10/bier_gesund.jpg
• http://t.qkme.me/3r3it3.jpg
• Intro_to_OWASP_Rochester_v5.ppt
• Owasp.org
• https://www.google.de/search?q=owasp&source=lnms&tbm=isch&sa=X