Cyber Security and Data Protection @ SoftDevHouse AG

OWASP Top 10

Niels Humbeck CSO (Chief Security Officer)

25.09.2021 Köln

Agenda

1. OWASP TOP 10 – Background information
2. Security Risk Nr.1 Injection
3. Security Risk Nr. 2 Broken Authentication
4. Security Risk Nr. 3 Sensitive Data Exposure
5. Security Risk Nr. 4 XML External Entities (XXE)

OWASP – Open Web Application Security Project

Nonprofit foundation improving software security

What is OWASP about?

- "Open community dedicated to enabling organizations to develop, purchase, and maintain applications and APIs that can be trusted" (OWASP-B)
- 20 years of experience in software security
- Since 2004 incorporated as a non-profit charity
- Founded 2001 by Mark Curphey

Core Values	
Open	Innovative
Global	Integrity

Community-led open-source software projects

- Juice Shop: on the job security training
- ZAP: free web application scanner

Cyber security awareness and training

- Cyber Security conferences & presentation
- Security Testing guidelines & standards
- OWASP Top 10 Vulnerabilities

See OWASP-A 2017 & OWASP-B 2017

Security Risk Nr.1 Injection

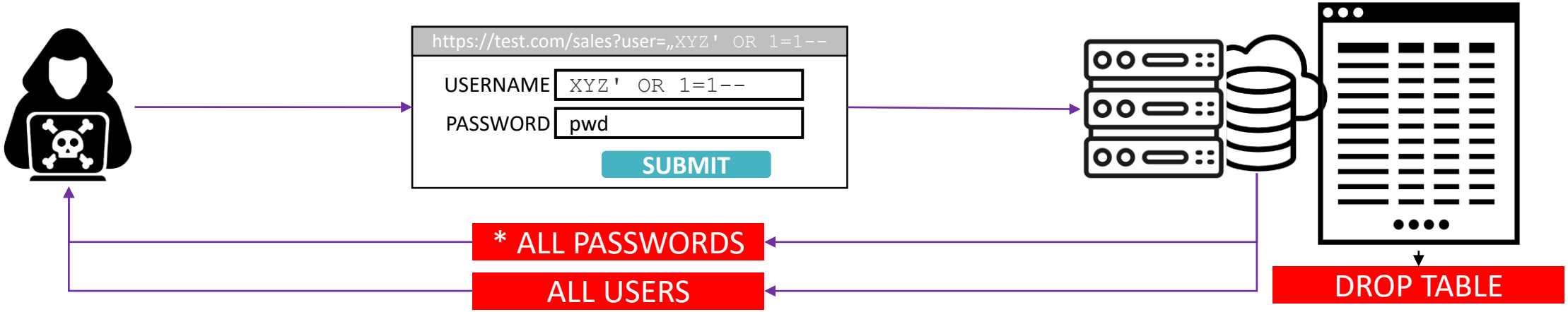
Use user inputs to sent malicious data to interpreter to manipulate a database

Attack vector	Prevalence	Detectability	Technical
All input fields interacting with a DB	Very prevalent	Easy to discover	Data loss, corruption, disclosure; misuse authentication details; examining the data base
3	2	3	3

```
XYZ' OR 1=1--
```

```
SELECT * FROM login WHERE user =
,XYZ' OR 1=1--' AND pwd = ,pwd'
```

Table with all sales data from every customer



Definition:

"Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization." [OWASP-B 2017]

OWASP-B (2017), OWASP-D (2021); OWASP-C (2021), Portswigger-A (2021), Portswigger-B (2021)

Security Risk Nr.1 Injection

How to prevent injections?

Input validation + limit queries

- Counteract any commands inserted in the input string (e.g. no quotation marks)
- E.g. postal zip code is a number between 0 and 6 figures

Performing testing

- Using e.g. Burp-Scanner (Portswigger) to test all input field to check for vulnerabilities

Web application firewall

- Using defined customizable web security rules to detect attacks by monitoring suspicious pattern in web traffic

Enforcing least privelege on database

- “Every program and every user […] should operate using the least set of privileges necessary to complete the job” (Barnum, 2013)

Parameterized queries

- Define all SQL code before binding the parameter (user input)
- Database can distinguish between code and data

```
//Get the username from the input field customerName
String custname = request.getParameter("customerName");
// Perform input validation to detect attacks
String query = "SELECT quantity FROM sales_data WHERE user_name = ? ";
PreparedStatement pstmt = connection.prepareStatement( query );
pstmt.setString( 1, custname);
ResultSet results = pstmt.executeQuery( );
```

See OWASP-D (2021)

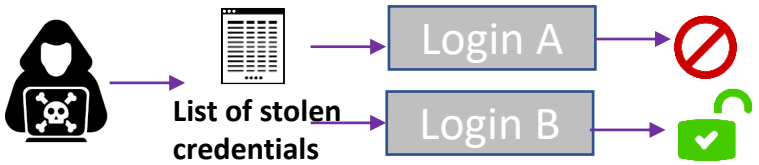
Sources: ptsecurity 2019, OWASP-C 2021 , Barnum (2013)

Security Risk Nr. 2 Broken Authentication

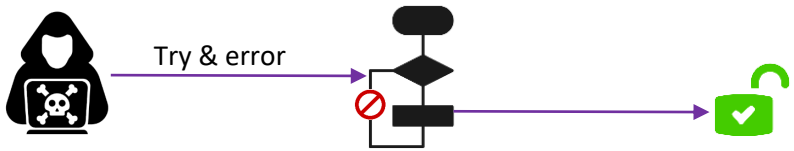
Gain unauthorized access by stealing user credentials or forge session data

Attack vector	Prevalence	Detectability	Technical
Credential leakages, Dictionary & brute force attack (3)	Widespread (2)	Lists of pwned pwds (2)	money laundering, social security fraud, identity theft, or disclose of sensitive information. (3)

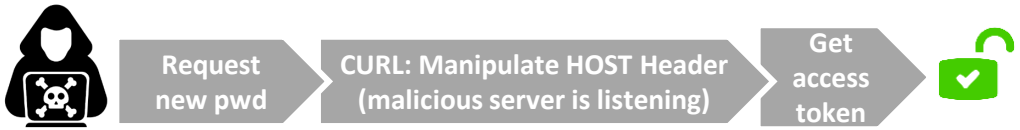
Credential stuffing



Brute force and weak pwds

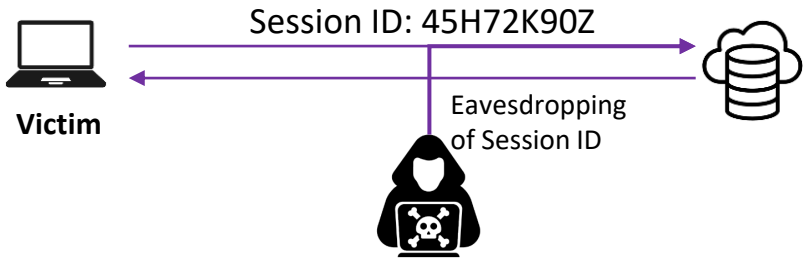


Weak credential recovery



With OWASP ZAP Proxy (Host Header injection)

Wrong Session Management



Definition:

"Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently." (OWASP-B 2017)

OWASP-H (2021), Thehackish (2020)

Security Risk Nr. 2 Broken Authentication

How to prevent Broken Authentification and poor Session Management

Attack vector	Prevalence	Detectability	Technical
Username + pwd leakages, Dictionary & brute force attack (3)	Widespread (2)	Manual means, pwd lists etc. (2)	money laundering, social security fraud, identity theft, or disclose of sensitive information. (3)

Credential stuffing, Brute force attacks and weak pwds

- Multi-Factor Authentication
- Requirements for strong pwds
- Identification of leaked pwd → alert
- Notification of unusual security events

Weak credential recovery

- Consistent message and time respond for (non)- & existing user + CAPTCHA
- Use URL token:
  - Tokens are randomly & cryptographically safe generated
  - Sent this token via mail
  - Don't rely on HOST header while creating the URL (prevent Host Header Injection)

Wrong Session Management

- Up to date framework for session identifier token generation & management
- Ensure the confidentiality of the session identifier (never expose the session identifier in the URL)
- Identifier token has a browser session lifetime with an inactivity timeout + termination by logout

Thehackish (2020) OWASP-F (2021); Affinity (2021), OWASP-G (2021)

Security Risk Nr. 3 Sensitive Data Exposure

Sensitive data exposure is mostly secondary to another type of attack

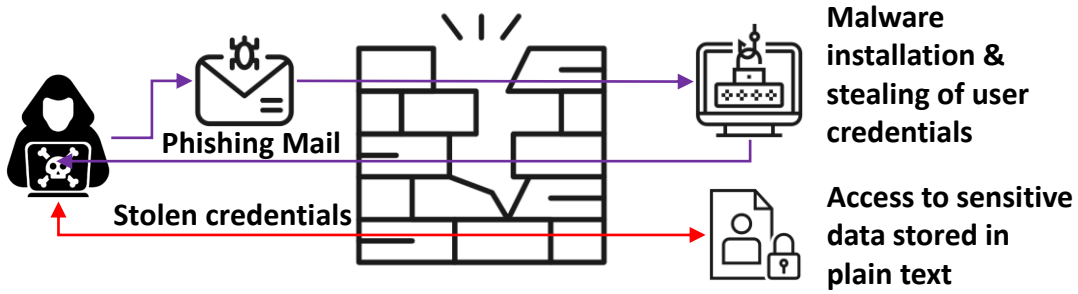
Attack vector 2
Credential stealing or man in the middle attack, etc.

Prevalence 3
Most common impactful attack

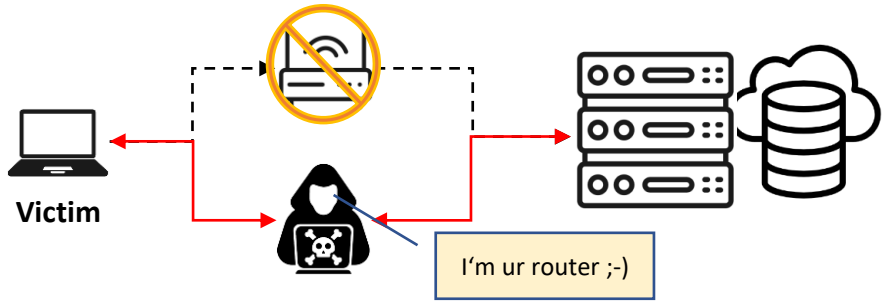
Detectability 2
Easy for data in motion, hard for data at rest

Technical 3
Compromises all "protected" data: sensitive personal information (Health care, credit card etc.)

Data at Rest



Data in Motion



- Ettercap is poisoning the ARP Cache
- Wireshark is reading the packages

Definition:

"Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser." (OWASP-B 2017)

Kudkar, I. (2021), SimplyExplained (2018)

Security Risk Nr. 3 Sensitive Data Exposure

Prevention sensitive data exposure

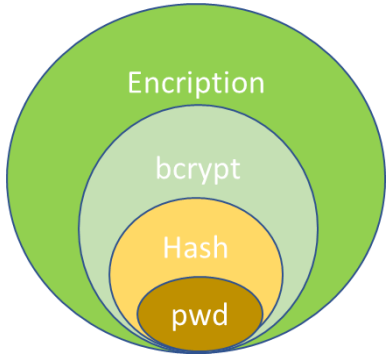
General (see OWASP-B):

- Classify data processed, stored, or transmitted by an application.
- Apply controls as per the classification.
- Verify independently the effectiveness of configuration and settings.
- Ensure up-to-date and strong standard algorithms, protocols, and keys are in place; use proper key management.
- Discard unnecessary sensible data ASAP

Data at Rest

Best practice for storage of sensible data (Docker):

- Storage of sensitive data:
  - Hashed
  - Salted hashed (e.g. bcrypt)
  - Encryption



Data in Motion (see OWASP-B)

- Use secure protocols such as TLS with forward secrecy (PFS) ciphers, cipher prioritization by the server, and secure parameters.
- Enforce encryption using directives like HTTP Strict Transport Security (HSTS).
- Disable caching for responses that contain sensitive data.

OWASP-B (2017), Kudkar, I. (2021), SimplyExplained (2018)

Security Risk Nr. 4 XML External Entities (XXE)

Commonly used for exfiltration of data via vulnerable XML interpreter

Attack vector 2

Exploitation of vulnerable XML Interpreter

Prevalence 2

Many older XML processors allow ext. entities

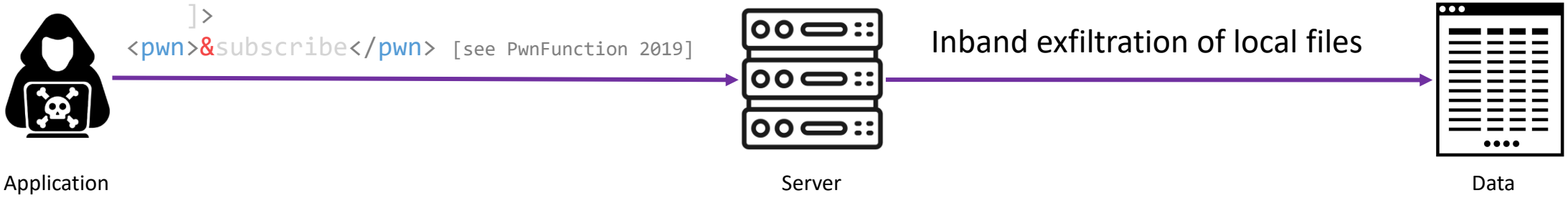
Detectability 3

Automated tests (SAST) and manual testing

Technical 3

Extract data, execute request, scan systems, DOS attacks

```
<?xml version="1.0"?>
<!DOCTYPE XXE [
    <!ENTITY subscribe SYSTEM "URL.txt">
    ]>
<pwn>&subscribe</pwn> [see PwnFunction 2019]
```



Definition:

"Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks." (OWASP-B 2017)

Portswigger-C (2021), OWASP-B (2017)

Prevention of XXE Attacks

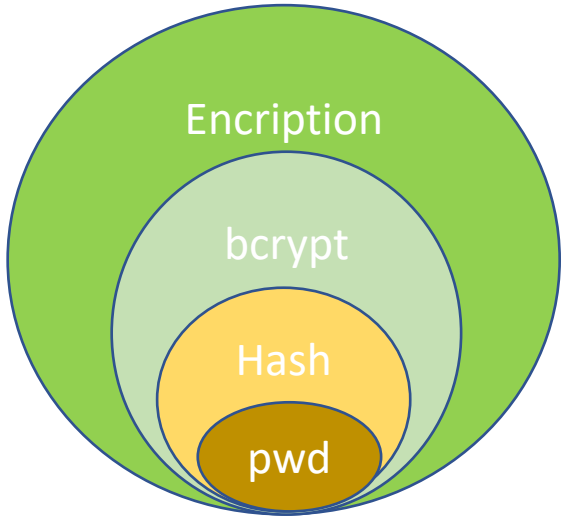
General (see OWASP-B 2017):

- Usage of fewer complex data formats like JSON
- Upgrading and Patching of XML-Interpreter (e.g. with dependencies checker)
- Disabling all external entities and DTD(Document Type Definition) in XML interpreter
- Use XML-Schema validation (XSD) for verifying uploaded XML files
- Automated testing/ scanning (e.g. with Burp scanner from Portswigger [Portswigger-C (2021)])
- Manual testing

QUESTIONS

Library - Icons

- https://owasp.org/ Icon
- https://thenounproject.com/term/arrow/729424/
- https://thenounproject.com/search/?q=arrow&i=147360
- https://thenounproject.com/search/?q=dissemination&i=1687523
- https://thenounproject.com/search/?q=dissemination&i=3967834
- https://thenounproject.com/search/?q=hacker&i=1531481
- https://thenounproject.com/search/?q=data+table&i=645988
- https://thenounproject.com/search/?q=harm&i=3338156
- https://thenounproject.com/search/?q=blocked&i=1957724
- https://thenounproject.com/search/?q=unlocked&i=975448
- https://thenounproject.com/search/?q=algorithm&i=1705449



- https://thenounproject.com/search/?q=server&i=4243541
- https://thenounproject.com/term/phishing/2466266/
- https://thenounproject.com/search/?q=www&i=3040077
- https://thenounproject.com/search/?q=router&i=4238531
- https://thenounproject.com/search/?q=smartphone&i=374036
- https://thenounproject.com/search/?q=laptop&i=1999
- https://thenounproject.com/term/firewall/3942098/

Library (1/2)

- Affinity (2021), „How to prevent session management vulnerabilities“, https://affinity-it-security.com/how-to-prevent-session-management-vulnerabilities/, last access 13.09.2021 at 17:12
- OWASP-A (2021), https://owasp.org/about/ last access at 22.08.2021 15:14
- OWASP-B, (2017), OWASP Top 10 2017: The ten most critical web applicatin security risks, https://owasp.org/www-pdf-archive/OWASP_Top_10-2017_%28en%29.pdf.pdf, 22.08.2021 16:00
- OWASP-C (2021) ; https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/05-Testing_for_SQL_Injection, last access 11.09.2021 at 07:00
- OWASP-D (2021); „SQL Injection Prevention Cheat Sheet“, https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html, last access 11.09.2021 at 07:31
- OWASP-E (2021);“Session Management Cheat Sheet¶; https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html, at 13.09.2021 at 16:21
- OWASP-F (2021); „Credential Stuffing Prevention Cheat Sheet¶“; https://cheatsheetseries.owasp.org/cheatsheets/Credential_Stuffing_Prevention_Cheat_Sheet.html, last access at13.09.2021 at 16:38
- OWASP-G (2021); „Forgot Password Cheat Sheet “; https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html, last access at 13.09 at 16:45
- OWASP-H (2021); „Testing for Host Header Injection“; https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/07-Input_Validation_Testing/17-Testing_for_Host_Header_Injection, last access 13.09.2021 at 16:58

Library (2/2)

- Portswigger-A (2021), „SQL Injection“, https://portswigger.net/web-security/sql-injection, last access 11.09.2021 at 06:44
- Portswigger-B (2021),“ Using Burp to Detect SQL Injection Flaws“, https://portswigger.net/support/using-burp-to-detect-sql-injection-flaws , last access 11.09.2021 at 07:05
- Portswigger-C (2021), „XML external entity (XXE) inkection“, https://portswigger.net/web-security/xxe, last access 19.09.2021 at 13:57
- Ptsecurity (2019),“How to prevent SQL injection attacks“, https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-prevent-sql-injection-attacks/, last access 11.09.2021 at 06:46
- Barnum, M. (2013),“Least Privilege“, https://us-cert.cisa.gov/bsi/articles/knowledge/principles/least-privilege, last access 11.09.2021 at 07:11
- Thehackish (2020), Hands-on OWASP Broken Authentication and Session Management tutorial - OWASP Top 10 training series, https://www.youtube.com/watch?v=l0_LtN_g6vg , last access 11.092021 at 09:45
- SimplyExplained (2018), „Passwords & hash functions (Simply Explained)“, https://www.youtube.com/watch?v=cczlpiiu42M, last access 14.09.2021 at 20:55
- PwnFunction (2019), XML External Entities (XXE) Explained, https://www.youtube.com/watch?v=gjm6VHZa_8s, last access 19.09.2021 at 14:00
- Kudkar, I. (2021), „OWASP: Senstive Data Exposure Attack“, Medium