

Bug Bounty Crash Course

Web Application Security Edition
Day 5

OWASP Top 10 : A1 Injection

A1
:2017

7

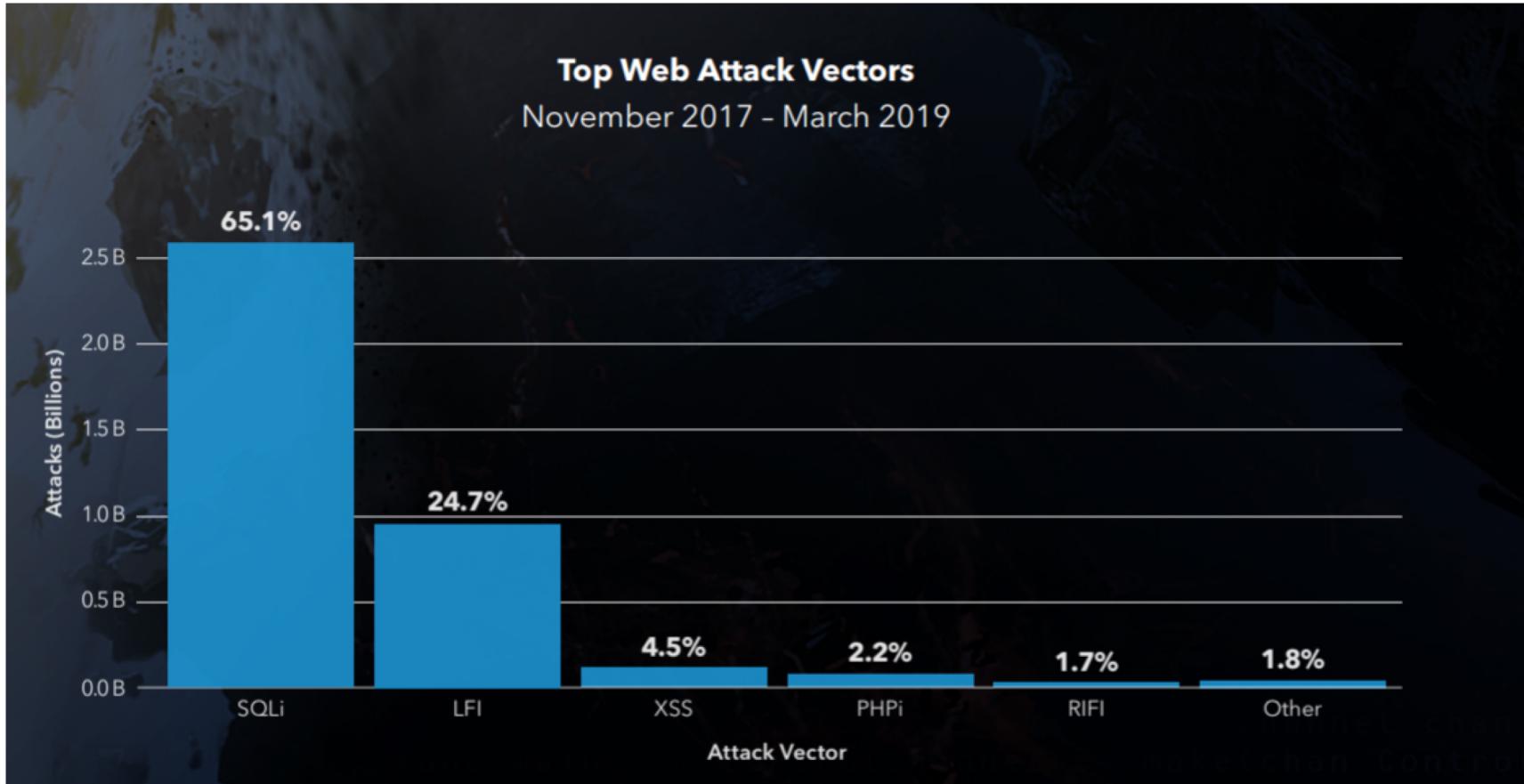
Injection

Threat Agents	Attack Vectors	Security Weakness	Impacts		
App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 3	Technical: 3	Business ?
Almost any source of data can be an injection vector, environment variables, parameters, external and internal web services, and all types of users. Injection flaws occur when an attacker can send hostile data to an interpreter.	Injection flaws are very prevalent, particularly in legacy code. Injection vulnerabilities are often found in SQL, LDAP, XPath, or NoSQL queries, OS commands, XML parsers, SMTP headers, expression languages, and ORM queries. Injection flaws are easy to discover when examining code. Scanners and fuzzers can help attackers find injection flaws.	Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. Injection can sometimes lead to complete host takeover. The business impact depends on the needs of the application and data.			

Source: OWASP

©PentesterAcademy.com

OWASP Top 10 : A1 Injection



Source: <https://www.cronline.com/news/sql-injection-attacks>

©PentesterAcademy.com



OWASP Top 10 : A1 Injection

Sophos XG Firewall
0day vulnerability
gets patched

27 April 2020



Hacker breached 60+ unis, govt agencies via SQL injection

A hacker tied to the November 2016 penetration of the US Election Assistance Commission and subsequent database sale has successfully targeted 60+ government agencies and universities by leveraging the same attack method: SQL injection.

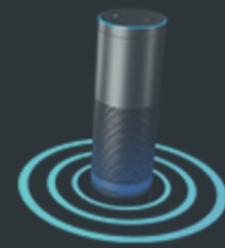
SQL Injection Attacks Represent Two-Third of All Web App Attacks

When Local File Inclusion attacks are counted, nearly nine in 10 attacks are related to input validation failures, Akamai report shows.

'Alexa, hack my serverless technology' – attacking web apps with voice commands

Amazon's voice assistant wisecracks her way through SQL injection attacks

11 December 2019

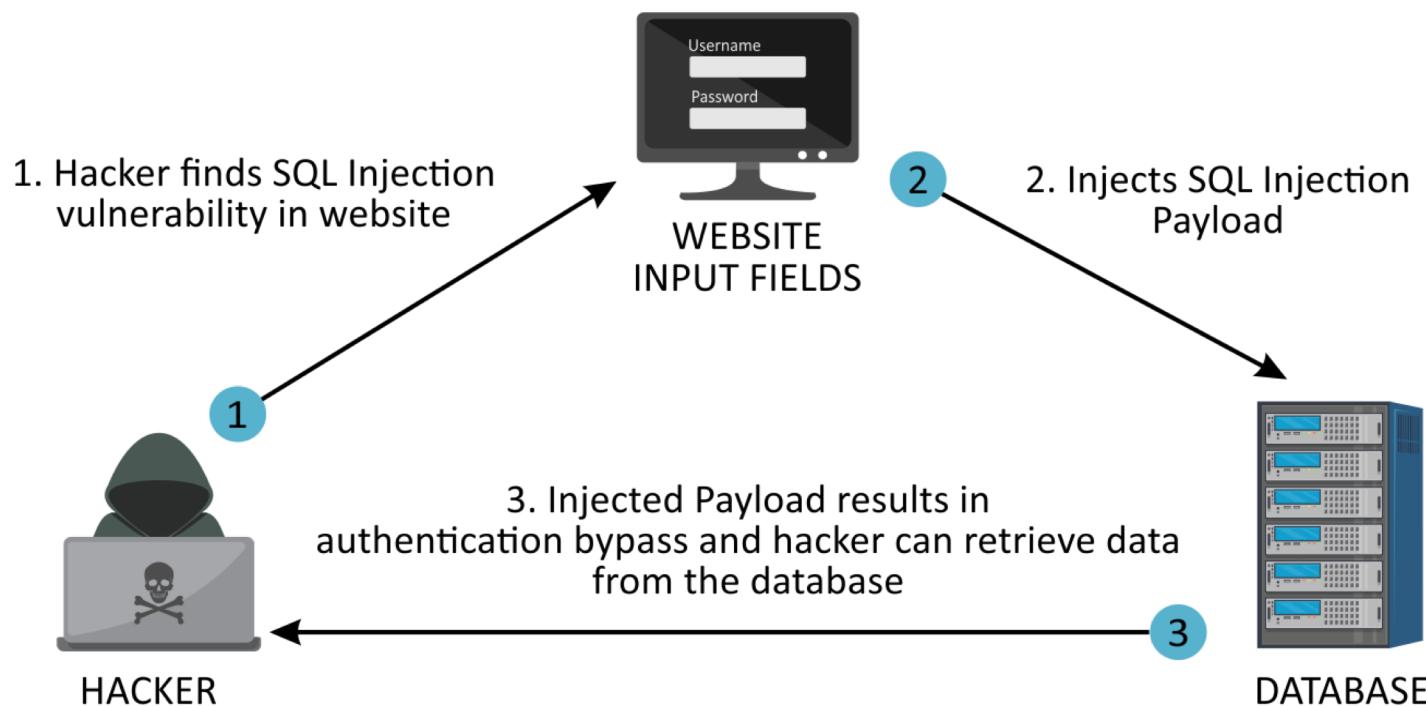


SQL injection flaw opened doorway to Starbucks' database

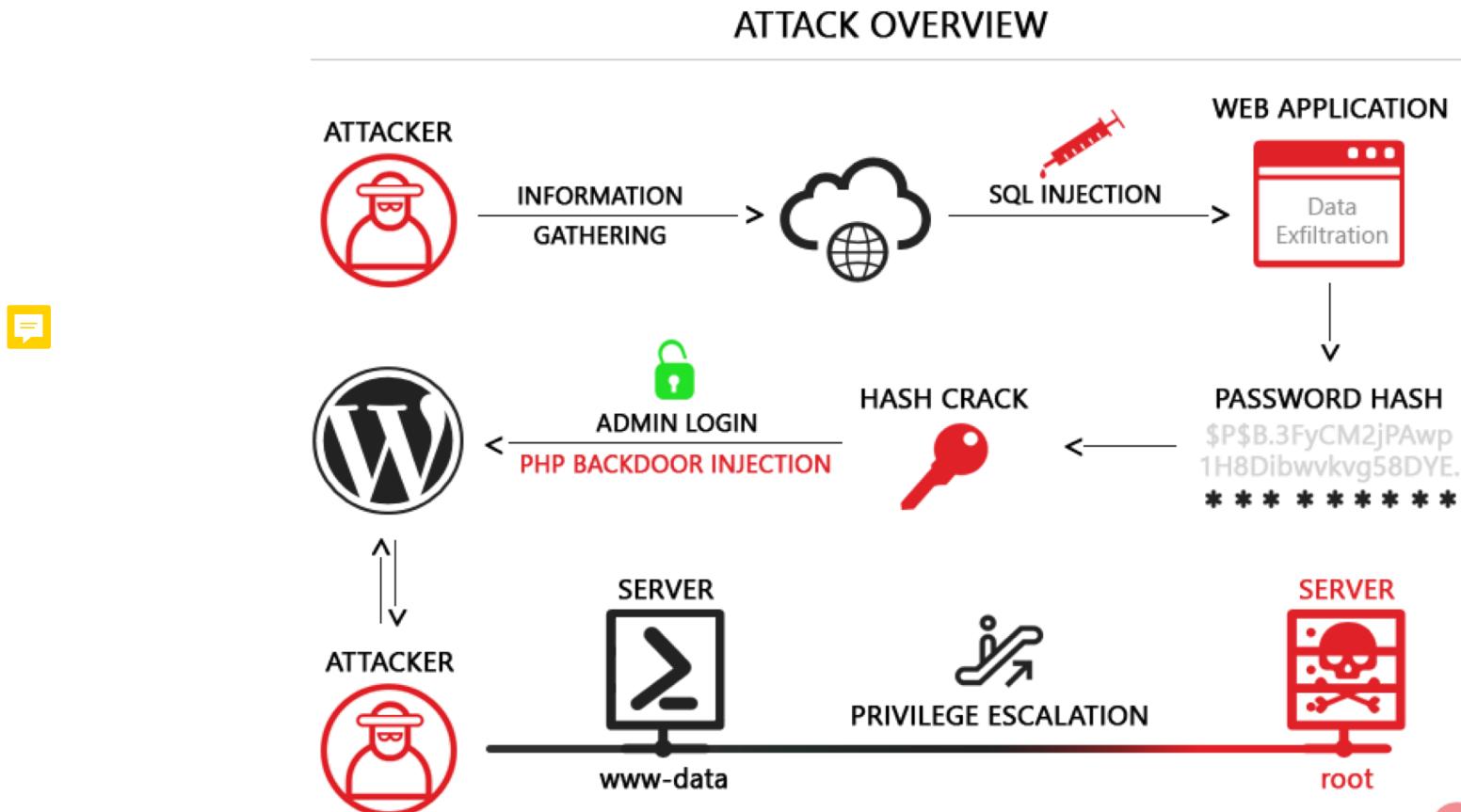
08 August 2019

SQL Injection

SQL Injection Attack (SQLi)



Total Compromise

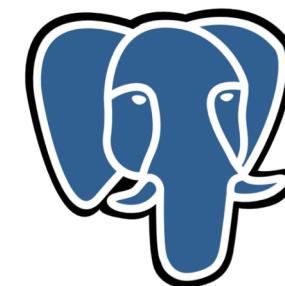


Source: <https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/>

©PentesterAcademy.com

Databases – SQL Based

- Structured storage and query of data
- Relational DB System - MySQL, PostgreSQL
- Embedded DB - SQLite



PostgreSQL

Lab: SQL Basics

<https://www.attackdefense.com/paredirect?cid=1801>

Authentication Bypass

SQL Injection



WEB PAGE

USERNAME: John
PASSWORD: *****

```
select * from users where
username='john' and password ='doe'
```



WEB PAGE

USERNAME: 1' or '1'='1
PASSWORD: *****

```
select * from users where username='1'
or '1'='1' and password ='1' or '1'='1'
```

Authentication Bypass

- Identifying vulnerability
- Trying generic SQLI Payloads
 - ' or '1'='1
 - ' or '1'='1' -- (Comment)
 - ' or '1'='1' # (Comment)
- Resultant SQL Query:
 - select * from users where login=" or '1'='1' and password=" or '1'='1'; ← always True
 - select * from users where login=" or '1'='1' -- ' and password="; ← always True
 - select * from users where login=" or '1'='1' # ' and password="; ← always True



Authentication Bypass

- Checking for SQLI vulnerability by injecting "" payload in login text field:

The screenshot shows a web page titled "SQL Injection (Login Form/Hero)". The page has a light gray background with a white form area. At the top, it says "Enter your 'superhero' credentials." Below this are two input fields: "Login:" and "Password:", each with a corresponding empty rectangular input box. Below the password input is a "Login" button with a rounded rectangle outline. At the bottom of the page, there is an error message: "Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "" AND password = "" at line 1".

/ SQL Injection (Login Form/Hero) /

Enter your 'superhero' credentials.

Login:

Password:

Login

Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "" AND password = "" at line 1

Lab: Free Article Submissions

Lab URL: <https://attackdefense.com/challengedetails?cid=315>

Lab: OpenSupports

Lab URL: <https://attackdefense.com/challengedetails?cid=437>

Homework Lab: Basic SQL Injection

Lab URL: <https://attackdefense.com/challengedetails?cid=1901>

SQL Injection Types

- Classic SQL Injection
 - Union Based SQL Injection
 - Error Based SQL Injection
- Blind SQL Injection
 - Boolean Based
 - Time Based
- Out of bound SQLI

OWASP Top 10 : A2 Broken Authentication

A2
:2017

8

Broken Authentication

```
graph LR; TA[Threat Agents] --> AV[Attack Vectors]; AV --> SW[Security Weakness]; SW --> I[Impacts]
```

App. Specific	Exploitability: 3	Prevalence: 2	Detectability: 2	Technical: 3	Business ?
Attackers have access to hundreds of millions of valid username and password combinations for credential stuffing, default administrative account lists, automated brute force, and dictionary attack tools. Session management attacks are well understood, particularly in relation to unexpired session tokens.		The prevalence of broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the bedrock of authentication and access controls, and is present in all stateful applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.		Attackers have to gain access to only a few accounts, or just one admin account to compromise the system. Depending on the domain of the application, this may allow money laundering, social security fraud, and identity theft, or disclose legally protected highly sensitive information.	

Source: OWASP

©PentesterAcademy.com

OWASP Top 10 : A2 Broken Authentication

HTTP is a “stateless” protocol

- Means credentials have to go with every request
- Should use SSL for everything requiring authentication

Session management flaws

- SESSION ID used to track state since HTTP doesn't
 - and it is just as good as credentials to an attacker
- SESSION ID is typically exposed on the network, in browser, in logs, ...

Beware the side-doors

- Change my password, remember my password, forgot my password, secret question, logout, email address, etc...

Typical Impact

- User accounts compromised or user sessions hijacked

Source: OWASP

OWASP Top 10 : A2 Broken Authentication

- Improper Authentication ranked 13th on Mitre CWE list

Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10
[11]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	11.47
[12]	CWE-787	Out-of-bounds Write	11.08
[13]	CWE-287	Improper Authentication	10.78
[14]	CWE-476	NULL Pointer Dereference	9.74

OWASP Top 10 : A2 Broken Authentication

```
import com.imperva.apiaattacktool.model.value.EndpointValueModel;
import com.imperva.apiaattacktool.model.value.factory.Property;
import com.imperva.apispecparser.model.EndpointModel;

import java.util.Collections;
```

Hostile takeover

Cybercriminals hammer APIs in a bid to bypass stronger authentication

20 February 2020

```
@Override
public List<EndpointValueModel> endpointModelToE(
    List<EndpointValueModel> endpointValueModel
) {
    if (endpointModelList != null) {
        endpointValueModelList = endpointModelList.stream()
```

Critical Authentication Bypass Vulnerability in InfiniteWP Client Plugin

This entry was posted in [Vulnerabilities](#), [WordPress Security](#) on January 14, 2020 by [Matt Barry](#) [8 Replies](#)

thehackernews.com › computer security › hacking ▾

[Authentication Bypass Vulnerability Found in Auth0 Identity ...](#)

Apr 7, 2018 - A critical **authentication bypass vulnerability** has been discovered in ... due to improper validation of the JSON Web Tokens (JWT) audience ...



©PentesterAcademy.com



Terminologies

- Authentication
 - the process or action of verifying the identity of a user or process
- Authorization
 - security mechanism used to determine user/client privileges or access levels related to system resources

When is the application vulnerable?

- Permits Automated attacks such as credential stuffing
- Permits brute force or other automated attacks
- Permits default, weak or well-known passwords
- Uses Weak or ineffective credential recovery and forgot-password processes
- Uses plain-text, encrypted or weakly hashed password
- Has missing or ineffective Multi-factor authentication

When is the application vulnerable?

- Exposes Session ID in URL
- Does not rotate session id after successful login
- Does not properly invalidate session ID

Installing Burp Suite and FoxyProxy

Video URL: <https://youtu.be/-JLUw3fr-ro>

Lab: Online Airline Booking System

Lab URL: <https://attackdefense.com/challengedetails?cid=438>

Improper Session Management

- bWAPP - Session Management - Administrative Portals
 - Admin session control parameter is URL

The screenshot shows a browser window for the bWAPP application. The URL in the address bar is `192.174.149.5/smgmt_admin_portal.php?admin=0`. The page has a yellow header with the bWAPP logo and a bee icon. Below the header, it says "an extremely buggy web app!". On the right side of the header, there's a "Choose your bug:" dropdown set to "bWAPP v2.2 -". Below that is a "Set your security level" dropdown set to "low". The main menu at the bottom includes links for Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, Logout, and a red "Welcome Bee". A red hand-drawn style text overlay at the top left of the page reads "/ Session Mgmt. - Administrative Portals /". Below this, a message says "This page is locked." and "HINT: check the URL...". To the right of the page, there are social media sharing icons for Twitter, LinkedIn, and Facebook.

Lab: Improper Session Management

Lab URL: <https://attackdefense.com/challengedetails?cid=1899>

Video: https://youtu.be/Yt-5Fgg-u_4

Improper Session Management - Cookie based

- bWAPP - Session Management - Administrative Portals
 - Admin Cookie

The screenshot shows the OWASP ZAP interface in proxy mode. The top navigation bar has tabs for Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Project options, and User options. Below the tabs, there are sub-tabs: Intercept (highlighted in red), HTTP history, WebSockets history, and Options. The main area shows a request to http://192.174.149.5:80. Below the request details are buttons for Forward, Drop, Intercept is on (which is on), and Action. At the bottom are tabs for Raw, Params, Headers, and Hex. The raw request text is as follows:

```
1 GET /smgmt_admin_portal.php HTTP/1.1
2 Host: 192.174.149.5
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: security_level=1; PHPSESSID=6prq52vmd3o4i6oq944l4rb392; admin=0
9 Upgrade-Insecure-Requests: 1
10
11
```

Lab: Improper Session Management II

Lab URL: <https://attackdefense.com/challengedetails?cid=1899>

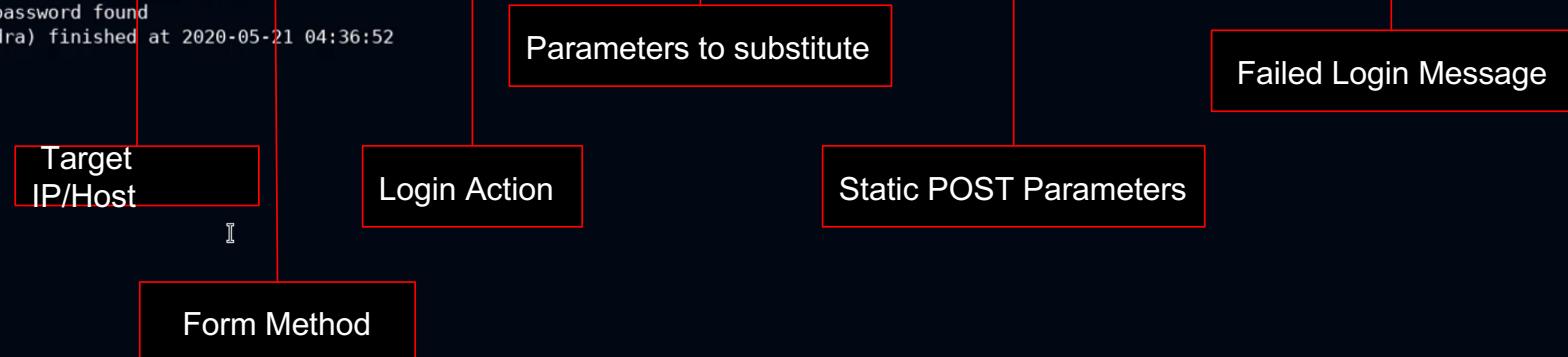
Video: <https://youtu.be/BvrKOK9Z7ok>

Weak Password

- bWAPP - Broken Authentication - Password Attacks
- Default passwords / Easy to remember
- Might be present in popular wordlists
- Depending Upon length, password can be brute forced.

HTTP Login Form : Hydra

```
root@attackdefense:~# hydra -L usernames -P passwords 192.195.214.3 http-post-form "/login.php:login^USER^&password^PASS^&security_level=0&form=submit:Invalid credentials or user not activated!"  
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-05-21 04:36:48  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 202 login tries (l:2/p:101), ~13 tries per task  
[DATA] attacking http-post-form://192.195.214.3:80/login.php:login^USER^&password^PASS^&security_level=0&form=submit:Invalid credentials or user not activated!  
[80][http-post-form] host: 192.195.214.3 login: bee password: bug  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-05-21 04:36:52  
root@attackdefense:~#  
root@attackdefense:~#  
root@attackdefense:~#  
root@attackdefense:~#
```



Lab: Attacking HTTP Login Form with Hydra

Lab URL: <https://attackdefense.com/challengedetails?cid=1895>

Prevention

- Where possible, implement multi-factor authentication to prevent automated, credential stuffing and dictionary/brute force.
- Do not ship or deploy with any default credentials, particularly for admin users.
- Implement weak-password checks, such as testing new or changed passwords against a list of the top 10000 worst passwords
- Align password length, complexity and rotation policies with NIST 800-63 B's guidelines in section 5.1.1 for Memorized Secrets or other modern, evidence based password policies.

Source: OWASP

Prevention

- Ensure registration, credential recovery, and API pathways are hardened against account enumeration attacks by using the same messages for all outcomes.
- Limit or increasingly delay failed login attempts. Log all failures and alert administrators when credential stuffing, brute force, or other attacks are detected.
- 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.

Source: OWASP

OWASP Top 10 : A3 Sensitive Data Exposure

- Information Exposure ranked 4th on Mitre CWE list

Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10
[11]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	11.47
[12]	CWE-787	Out-of-bounds Write	11.08
[13]	CWE-287	Improper Authentication	10.78
[14]	CWE-476	NULL Pointer Dereference	9.74

OWASP Top 10 : A3 Sensitive Data Exposure

Ingram data breach

Digital content platform hack resulted in theft of publishers' titles

17 April 2020



ATTACKS/BREACHES

3/13/2020
12:15 PM

Princess Cruises Confirms Data Breach

The cruise liner, forced to shut down operations due to coronavirus, says the incident may have compromised passengers' personal data.

Carnival-owned Princess Cruises — the cruise line forced to suspend operations after two ships became hot spots for coronavirus — reports that a breach may have compromised passenger data.

DARK
Reading

Dark Reading Staff
Quick Hits

SFO data breach

Double website hack may have lifted users' Windows login credentials

14 April 2020



[threatpost.com](#) › Breach ▾

Millions of Guests Impacted in Marriott Data Breach, Again ...

Mar 31, 2020 - In 2019, the Information Commissioner's Office (ICO), which is the U.K.'s privacy watchdog, hit **Marriott** with a \$123 million (£99 million) penalty ...

OWASP Top 10 : A3 Sensitive Data Exposure

A3
:2017

9

Sensitive Data Exposure

Threat Agents	Attack Vectors	Security Weakness	Impacts		
App. Specific	Exploitability: 2	Prevalence: 3	Detectability: 2	Technical: 3	Business ?
Rather than directly attacking crypto, attackers steal keys, execute man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user's client, e.g. browser. A manual attack is generally required. Previously retrieved password databases could be brute forced by Graphics Processing Units (GPUs).	Over the last few years, this has been the most common impactful attack. The most common flaw is simply not encrypting sensitive data. When crypto is employed, weak key generation and management, and weak algorithm, protocol and cipher usage is common, particularly for weak password hashing storage techniques. For data in transit, server side weaknesses are mainly easy to detect, but hard for data at rest.	Failure frequently compromises all data that should have been protected. Typically, this information includes sensitive personal information (PII) data such as health records, credentials, personal data, and credit cards, which often require protection as defined by laws or regulations such as the EU GDPR or local privacy laws.			

Source: OWASP

©PentesterAcademy.com

When is the application vulnerable?

- Is any data transmitted in clear text? This concerns protocols such as HTTP, SMTP, and FTP. External internet traffic is especially dangerous
- Is sensitive data stored in clear text, including backups?
- Are any old or weak cryptographic algorithms used either by default or in older code?
- Are default crypto keys in use, weak crypto keys generated or re-used, or is proper key management or rotation missing?

Source: OWASP

When is the application vulnerable?

- Is encryption not enforced, e.g. are any user agent (browser) security directives or headers missing?
- Does the user agent (e.g. app, mail client) not verify if the received server certificate is valid?

Source: OWASP

©PentesterAcademy.com

Types of Storages

	Cookies	Local Storage	Session Storage
Capacity	4KB	10MB	5MB
Browsers	HTML4 / HTML 5	HTML 5	HTML 5
Accessible from	Any window	Any window	Same tab
Expires	Manually set	Never	On tab close
Storage Location	Browser and server	Browser only	Browser only
Sent with requests	Yes	No	No



Encoded Cookie value

- bWAPP - Base64 Encoding (Secret)

The screenshot shows a browser window for 'bWAPP - Sensitive Data' at the URL '192.174.149.5/insecure_crypt_storage_3.php'. The page has a yellow header with the bWAPP logo and a bee icon. Below it, the text 'an extremely buggy web app!' is displayed in red. On the right side of the header, there's a sidebar with 'Choose your bug:' and a dropdown menu set to 'bWAPP'. Below that is a section for 'Set your security level' with a dropdown set to 'low'. The main content area has a title '/ Base64 Encoding (Secret) /'. It says 'Your secret has been stored as an encrypted cookie!' and 'HINT: try to decrypt it...'. To the right of the content are social media sharing icons for Twitter, LinkedIn, and Facebook.



Lab: Encoded Cookie Value

Lab URL: <https://attackdefense.com/challengedetails?cid=1899>

Video URL: <https://youtu.be/NnweEYixzQg>

Data in Local Storage

- bWAPP - Base64 Encoding (Secret)

The screenshot shows a browser window for the bWAPP application at the URL `192.174.149.5/insecure_crypt_storage_3.php`. The page has a yellow header with the bWAPP logo and a bee icon. Below the header, the text "an extremely buggy web app!" is displayed in red. On the right side of the header, there is a sidebar with the heading "Choose your bug:" followed by a dropdown menu set to "bWAPP". Below this, another section is labeled "Set your security level" with a dropdown menu set to "low". At the bottom of the page, there is a navigation bar with links: "Bugs", "Change Password", "Create User", "Set Security Level", "Reset", "Credits", "Blog", "Logout", and "Welcome [username]". The main content area below the navigation bar contains the text "/ Base64 Encoding (Secret) /" in large, stylized letters. Underneath this, there is a message: "Your secret has been stored as an encrypted cookie!" and a hint: "HINT: try to decrypt it...". To the right of the main content, there are social media sharing icons for Twitter, LinkedIn, and Facebook.

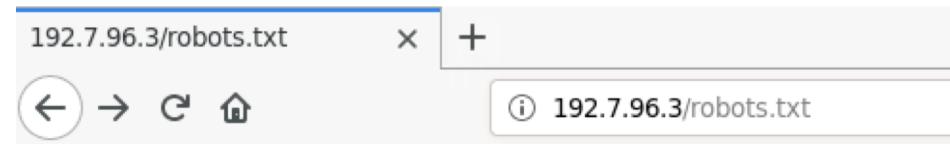
Lab: Sensitive Data in Web Storage

Lab URL: <https://attackdefense.com/challengedetails?cid=1899>

Video URL: https://youtu.be/Y_6bbhD8x3o

Robots.txt

- Tells Crawler allowed/disallowed pages/files/directories for crawling
- Used for Traffic Management
 - Prevents site from being overloaded by requests made from crawlers
- Preventing particular crawler from crawling pages.



```
User-agent: GoodBot
Disallow:

User-agent: BadBot
Disallow: /

User-agent: *
Disallow: /admin/
Disallow: /documents/
Disallow: /images/
Disallow: /passwords/
```



Lab: Sensitive Directories in robots.txt

Lab URL: <https://attackdefense.com/challengedetails?cid=1889>

Video URL: <https://youtu.be/7s5RD4OHD4w>

Real World Webapp

◀ All Section Labs

CVE-2018-12604

cve-2018 | Level: **Easy** | Total Lab Runs: **1** | **Running**



Lab Link

Stop

Lab Scoreboard

287

Played on AD

0

Played by you

Mark Complete

Mission Hint 1 Prohibited Activities Technical Support

Download Lab Manual

The attacker might not have any user level access to the web application. However, this does not mean that the application cannot be attacked remotely. Sensitive Information Disclosure vulnerabilities could be triggered even by unauthenticated users.

In the exercise below, the attacker is unauthenticated to the web application and needs to find a sensitive information disclosure attack to access sensitive information from the server.

GreenCMS is a free and open source CMS developed in PHP.

GreenCMS (2.3.0603) is vulnerable to a Sensitive Information Disclosure documented in **CVE-2018-12604**.

Objective: Your task is to find and exploit this vulnerability.

CVE is a registered trademark of The MITRE Corporation.

Lab Link: <https://attackdefense.com/challengedetails?cid=11>

©PentesterAcademy.com

Lab: CVE-2018-12604

Lab URL: <https://attackdefense.com/challengedetails?cid=11>

Prevention

- Classify data processed, stored, or transmitted by an application.
Identify which data is sensitive according to privacy laws, regulatory requirements, or business needs
- Apply controls as per the classification
- Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation.
Data that is not retained cannot be stolen.
- Make sure to encrypt all sensitive data at rest. 

Source: OWASP