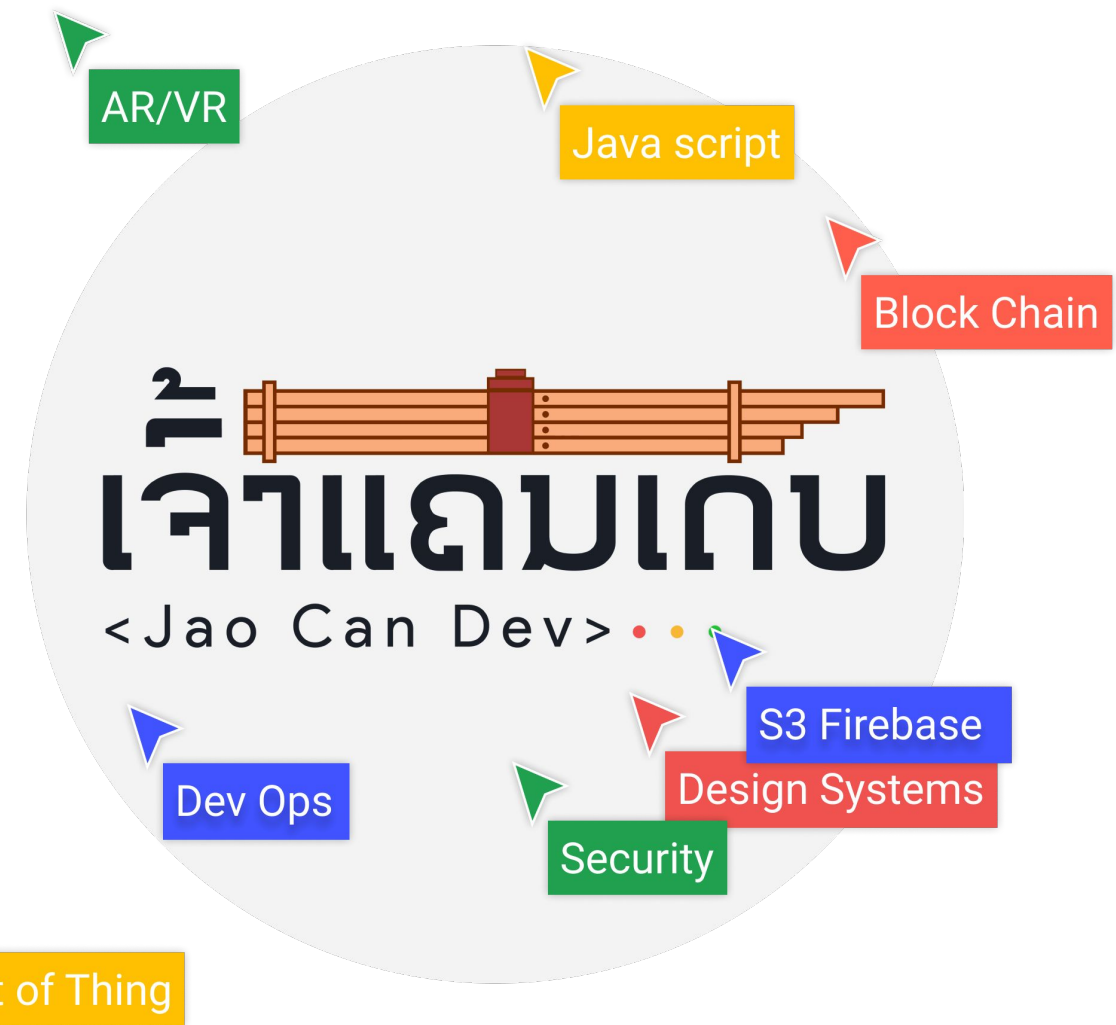




Secure Coding

ການຂຽນ Code ໃຫ້ປອດໄພ



Outhai SAIUDOM, Lao IT Dev Co.,Ltd, H.O.D Media, CQ Co.,Ltd

  @Os555

~# whoami

Name: Outhai SAIOUDOM

Nick: Os, (LoungOS)

Codename: Os555

Skills: Linux, CyberSec, NooB Cryptocurrency Trader

Position: Co-Founder / C.T.O - LaoITDev Co., Ltd.



Os555



ລຸງໂອ້ດ



Os555

LAOITDEV



LAO





**CUSTOMIZED
SOFTWARE
DEVELOPMENT**



**WEB DESIGN AND
DEVELOPMENT**



**MOBILE
APPLICATION
DEVELOPMENT**



**RFID AND IOT
SOLUTION**



**PENETRATION
TESTING AND
CYBER SECURITY**



**IT TRAINING
CENTER**

Secure Coding

ການຂຽນ Code ໃຫ້ປອດໄພ



ໃນໂລກນີ້ຈະມີຄົນຢູ່ 2 ປະເພດຄື:

1. ຄົນທີ່ຖືກແຮ້ກ

2. ຄົນທີ່ຖືກແຮ້ກແຕ່ຍັງບໍ່ຮູ້ໂຕວ່າຖືກແຮ້ກ



BRINGING CIVILIZATION TO ITS KNEES...

Goths



Vandals



Huns



Geeks



THE WORLD NEWS
ONE SOURCE FOR HEADLINES \$1.00

Since 1883

WORK
YOUR NUMBER ONE SOURCE FOR HEADLINES

\$1.00

1883 YOUR NUMBER ONE

CYBER ATTACK!

OF ATTACKS

DEFENSE DEPARTMENT
TO BEEF UP VIRUS
PROTECTION

**24,000 FILES STOLEN IN
LATEST ATTACK**

Record attacks are now being reported throughout the country and lower throughout the country are being

IMPACT OF ATTACKS REACHING INTO THE ECONOMY

Efforts are being made by local law enforcement industry leaders in order to impact pending in time for the consumers have

**DEFENSE DEPARTMENT
TO BEEF UP VIRUS
PROTECTION**

For updated information please visit our afternoon edition Water consumption up throughout the country. Efforts are being made by local community leaders in order to spending in time for the Consumers



THAILAND > GENERAL

Data of TrueMove H users leaked online

PUBLISHED : 15 APR 2018 AT 05:00

NEWSPAPER SECTION: NEWS

WRITER: SUCHIT LEESA-NGUANSUK AND KOMSAN TORTERMVASANA



The personal data of around 46,000 TrueMove H users was leaked into Amazon Web Services' cloud storage, leading the National Broadcasting and Telecommunications Commission to call in the company for questioning. (Photo by Narupon Hinshiranan)

The personal data of around 46,000 TrueMove H users was leaked into Amazon Web Services' (AWS) cloud storage, leading the National Broadcasting and Telecommunications Commission (NBTC) to call in the company for questioning on Saturday.

3BB ถูกแฮกข้อมูลลูกค้ากว่า 10,000 ราย

Posted on 15 Jan 2021 News

HOME

สมัครบริการ

ชำระค่าบริการ

3BB Member

3BB Product

3BB Shop

3BB Privilege

ช่องทางการชำระเงิน

ไทย EN

ลงชื่อเข้าสู่ระบบ



เรียนลูกค้า 3BB ทุกท่าน

กรณีมีกลุ่มแฮกเกอร์ (Hacker) ได้เผยแพร่ว่ามีการเข้าถึงข้อมูลภายในของกลุ่ม JAS และบริษัททริปเปิ้ลที บรอดแบนด์ จำกัด (มหาชน) หรือ 3BB และผู้กระทำความผิดได้เรียกเงินค่าไถ่แลกกับการไม่เปิดเผยข้อมูลดังกล่าวสู่สาธารณะ

บริษัทฯ ขอเรียนให้ทราบว่าในช่วงปลายปี 2563 ที่ผ่านมา ได้มีเหตุการณ์ที่เว็บไซต์ทั่วโลกได้ถูกกลุ่มแฮกเกอร์ทำการละเมิดโดยการเข้าถึงและนำข้อมูลออกไปจากเว็บไซต์ต่างๆ โดยมีขอบ เพื่อทำการเรียกเครื่องผลประโยชน์ ซึ่งในช่วงเวลาดังกล่าว บริษัทฯ ได้มีการเฝ้าระวังและพบว่ามีความพยายามเข้าถึงข้อมูลของบริษัทฯ อย่างผิดปกติ ทางบริษัทฯ จึงได้ดำเนินการปิดกั้นการเข้าถึงดังกล่าวทันทีและมีการเฝ้าระวังตลอดเวลา จากการตรวจสอบเบื้องต้นพบว่าข้อมูลลูกค้าบางส่วนของ 3BB Member ประมาณ 10,000 รายได้ถูกดึงไป เช่น ชื่อ-ที่อยู่ เบอร์โทรศัพท์ ข้อมูลวันเกิด หมายเลขบัตรประชาชน สำหรับรหัสผ่านนั้นระบบได้เข้ารหัสรักษาความปลอดภัยไว้ ส่วนสำเนาบัตรประชาชนไม่ได้ถูกเข้าถึง นอกจากนี้ข้อมูลบัตรเครดิตและข้อมูลทางการเงินของลูกค้าก็ไม่ได้ถูกเข้าถึงเนื่องจากบริษัทไม่ได้เก็บข้อมูลดังกล่าวไว้ในระบบแต่อย่างใด

บริษัทฯ รู้สึกเสียใจเป็นอย่างยิ่งต่อเหตุการณ์ในครั้งนี้ซึ่งทำให้ลูกค้ามีความกังวลในเรื่องของความปลอดภัยของข้อมูลส่วนตัว ทั้งนี้ขอยืนยันว่าบริษัทฯ มีระบบป้องกันรักษาความปลอดภัย อุปกรณ์ Firewall ระบบ Anti Virus และมาตรการตรวจสอบเฝ้าระวังอย่างสม่ำเสมอ แต่ก็ยังเปิดโอกาสที่กลุ่มผู้ไม่หวังดีที่มีความเชี่ยวชาญหรือ Hacker จะประสบความสำเร็จในการโจมตีและเข้าถึงระบบข้อมูลโดยใช้เทคนิควิธีการต่างๆ บริษัทฯ ไม่ได้มองบ่นใจกับสิ่งที่เกิดขึ้น โดยได้ทำการปิดกั้นการเข้าถึงข้อมูลของลูกค้าทั้งหมดในทันทีที่ตรวจพบ และได้เพิ่มมาตรการปิดกั้นการเข้าถึงที่ผิดปกติจาก IP Address ต่างประเทศ รวมทั้งปรับปรุงระบบความปลอดภัยของข้อมูลให้สูงขึ้น มีการดำเนินการจัดหาซอฟต์แวร์และจ้างที่ปรึกษาระบบความปลอดภัยจากภายนอกมาตรวจสอบระบบ เพื่อเพิ่มความมั่นใจในการรักษาความปลอดภัยของข้อมูลส่วนตัว อย่างไรก็ตาม ลูกค้าสามารถเข้าไปเปลี่ยนรหัสผ่านที่เว็บไซต์ 3BB หรือแอปพลิเคชัน 3BB Member เพื่อเพิ่มความปลอดภัยในการใช้งานมากยิ่งขึ้น นอกจากนี้ฝ่ายกฎหมายของบริษัทฯ ได้ดำเนินการเข้าแจ้งความกับตำรวจ และประสานขอความร่วมมือจากกองบังคับการปราบปรามการกระทำความผิดเกี่ยวกับอาชญากรรมทางเทคโนโลยี (บก.ปอท.) เป็นที่เรียบร้อยแล้ว



3BB BROADBAND



3BB BROADBAND

เมื่อวันที่ 12 มกราคม 2564 ที่ผ่านมา บริษัททริปเปิ้ลที บรอดแบนด์ จำกัด (มหาชน) หรือ 3BB ประกาศผ่านหน้าเว็บไซต์ของตนเอง ว่ามีกลุ่มแฮกเกอร์ (Hacker) ได้เผยแพร่ว่ามีการเข้าถึงข้อมูลภายในของกลุ่ม JAS และบริษัททริปเปิ้ลที บรอดแบนด์ จำกัด (มหาชน) หรือ 3BB และผู้กระทำความผิดได้เรียกเงินค่าไถ่แลกกับการไม่เปิดเผยข้อมูลดังกล่าวสู่สาธารณะ

ข้อความที่ทางบริษัททริปเปิ้ลที บรอดแบนด์ จำกัด (มหาชน) หรือ 3BB ประกาศลงหน้าเว็บไซต์มีดังนี้

“บริษัทฯ ได้มีการเฝ้าระวังและพบว่ามีความพยายามเข้าถึงข้อมูลของบริษัทฯ อย่างผิดปกติ ทางบริษัทฯ จึงได้ดำเนินการปิดกั้นการเข้าถึงดังกล่าวทันทีและมีการเฝ้าระวังตลอดเวลา จากการตรวจสอบเบื้องต้นพบว่าข้อมูลลูกค้าบางส่วนของ 3BB Member ประมาณ 10,000 รายได้ถูกดึงไป เช่น ชื่อ-ที่อยู่ เบอร์โทรศัพท์ ข้อมูลวันเกิด หมายเลขบัตรประชาชน สำหรับรหัสผ่านนั้นระบบได้เข้ารหัสรักษาความปลอดภัยไว้ ส่วนสำเนาบัตรประชาชนไม่ได้ถูกเข้าถึง นอกจากนี้ข้อมูลบัตรเครดิตและข้อมูลทางการเงินของลูกค้าก็ไม่ได้ถูกเข้าถึงเนื่องจากบริษัทไม่ได้เก็บข้อมูลดังกล่าวไว้ในระบบแต่อย่างใด บริษัทฯ ได้ทำการปิดกั้นการเข้าถึงข้อมูลของลูกค้าทั้งหมดในทันทีที่ตรวจพบ และได้เพิ่มมาตรการการป้องกัน การปรับปรุงระบบความปลอดภัยของข้อมูลให้สูงขึ้น อย่างไรก็ตาม ลูกค้าสามารถเข้าไปเปลี่ยนรหัสผ่านที่เว็บไซต์ 3BB หรือแอปพลิเคชัน 3BB Member เพื่อเพิ่มความปลอดภัยในการใช้งานมากยิ่งขึ้น”

ขอบคุณที่มาจาก [3BB](#)

<https://www.bitdefender.co.th/post/3bb-10-000>

Case Study #1 Facebook Token

Facebook Security Breach Exposes Accounts of 50 Million Users



One of the challenges for Facebook's chief executive Mark Zuckerberg is convincing users that the company handles their data responsibly. Josh Edelson/Agence France-Presse — Getty Images

First: View As is a privacy feature that lets people see what their own profile looks like to someone else. View As should be a view-only interface. However, for one type of composer (the box that lets you post content to Facebook) — specifically the version that enables people to wish their friends happy birthday — View As incorrectly provided the opportunity to post a video.

Second: A new version of our video uploader (the interface that would be presented as a result of the first bug), introduced in July 2017, incorrectly generated an access token that had the permissions of the Facebook mobile app.

Third: When the video uploader appeared as part of View As, it generated the access token not for you as the viewer, but for the user that you were looking up.

It was the combination of these three bugs that became a vulnerability: when using the View As feature to view your profile as a friend, the code did not remove the composer that lets people wish you happy birthday; the video uploader would generate an access token when it shouldn't have; and when the access token was generated, it was not for you but the person being looked up. That access token was then available in the HTML of the page, which the attackers were able to extract and exploit to log in as another user.

OWASP Web Top 10 Application Security Risks - 2017

A1: Injection

**A2: Broken Authentication
and Session Management**

A3: Sensitive Data Exposure

**A4: XML External Entities
(XXE)**

A5: Broken Access Control

**A6: Security
Misconfiguration**

**A7: Cross-Site Scripting
(XSS)**

A8: Insecure Deserialization

**A9: Using Known Vulnerable
Components**

**A10: Insufficient Logging &
Monitoring**

OWASP Web Top 10 Application Security Risks - 2017

What is My Risk?

The [OWASP Top 10](#) focuses on identifying the most serious web application security risks for a broad array of organizations. For each of these risks, we provide generic information about likelihood and technical impact using the following simple ratings scheme, which is based on the [OWASP Risk Rating Methodology](#).

Threat Agents	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
App. Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	SEVERE: 3	App / Business Specific
	AVERAGE: 2	COMMON: 2	AVERAGE: 2	MODERATE: 2	
	DIFFICULT: 1	UNCOMMON: 1	DIFFICULT: 1	MINOR: 1	

A1: Injection



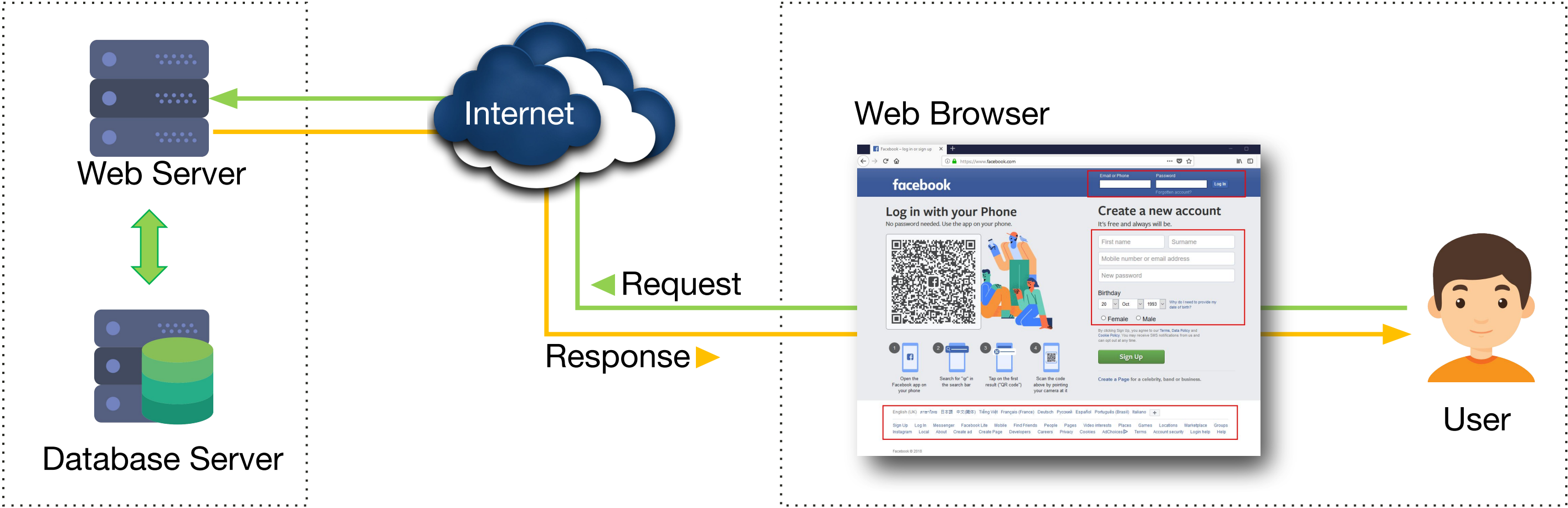
Input Validation

- ☐ **Overview**
- ☐ SQL Injection
- ☐ Command Injection
- ☐ Directory Traversal
- ☐ Blacklist vs. Whitelist validation
- ☐ Client Side vs Server Side Validation

How Does Web / Mobile Application Work

Server Side

Client Side





Input Validation

- ☐ Overview
- ☒ **SQL Injection**
- ☐ Command Injection
- ☐ Directory Traversal
- ☐ Blacklist vs. Whitelist validation
- ☐ Regular expressions(Regex)
- ☐ Client Side vs Server Side Validation

The core security problem is
“User can submit arbitrary input”

So....

Never Trust Client Input

What is an input?

Facebook – log in or sign up

https://www.facebook.com

facebook

Email or Phone


Password

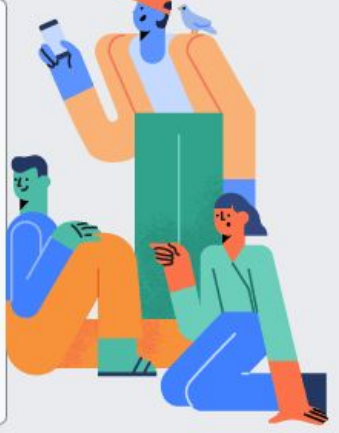
Log In

Forgotten account?


Log in with your Phone

No password needed. Use the app on your phone.






1




Open the Facebook app on your phone

2




Search for "qr" in the search bar

3



Tap on the first result ("QR code")

4



Scan the code above by pointing your camera at it

Create a new account

It's free and always will be.

First name

Surname

Mobile number or email address

New password

Birthday

20

Oct

1993

Why do I need to provide my date of birth?

☐ Female

☐ Male

By clicking Sign Up, you agree to our [Terms](#), [Data Policy](#) and [Cookie Policy](#). You may receive SMS notifications from us and can opt out at any time.

Sign Up

Create a Page for a celebrity, band or business.

English (UK) ไทย ภาษาไทย 日本語 中文(简体) Tiếng Việt Français (France) Deutsch Русский Español Português (Brasil) Italiano

+

Sign Up

Log In

Messenger

Facebook Lite

Mobile

Find Friends

People

Pages

Video interests

Places

Games

Locations

Marketplace

Groups

Instagram

Local

About

Create ad

Create Page

Developers

Careers

Privacy

Cookies

AdChoices

Terms

Account security

Login help

Help

Facebook © 2018

SQL Injection Example



Failed Code #1

```
1  <?php
2
3  if( isset( $_REQUEST[ 'Submit' ] ) ) {
4      // Get input
5      $id = $_REQUEST[ 'id' ];
6
7      // Check database
8      $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
9      $result = mysqli_query($GLOBALS["__mysqli_ston"], $query ) or die( '<pre>' . ((is_object($GLOBALS["__mysqli_ston"])) ?
      mysqli_error($GLOBALS["__mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ? $___mysqli_res : false)) . '</pre>' );
10
11     // Get results
12     while( $row = mysqli_fetch_assoc( $result ) ) {
13         // Get values
14         $first = $row["first_name"];
15         $last = $row["last_name"];
16
17         // Feedback for end user
18         echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
19     }
20
21     mysqli_close($GLOBALS["__mysqli_ston"]);
22 }
23
24 ?>
```


SQL Injection (Example)

```
$id = 1;  
$query = "SELECT first_name, last_name FROM users WHERE user_id = '1'; "  
  
SELECT first_name, last_name FROM users WHERE user_id = '1';
```

```
$id = 2;  
$query = "SELECT first_name, last_name FROM users WHERE user_id = '2'; "  
  
SELECT first_name, last_name FROM users WHERE user_id = '2';
```

```
$id = ' or '1'='1';  
$query = "SELECT first_name, last_name FROM users WHERE user_id = ' or '1'='1'; "  
  
SELECT first_name, last_name FROM users WHERE user_id = ' or '1'='1';
```

SQL Injection (Example)

```
SELECT * FROM users WHERE username = admin and password = ' or '1'='1';
```

Secure Code

```
1  <?php
2
3  if( isset( $_GET[ 'Submit' ] ) ) {
4      // Check Anti-CSRF token
5      checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );
6
7      // Get input
8      $id = $_GET[ 'id' ];
9
10     // Was a number entered?
11     if(is_numeric( $id )) {
12         // Check the database
13         $data = $db->prepare( 'SELECT first_name, last_name FROM users WHERE user_id = (:id) LIMIT 1;' );
14         $data->bindParam( ':id', $id, PDO::PARAM_INT );
15         $data->execute();
16         $row = $data->fetch();
17
18         // Make sure only 1 result is returned
19         if( $data->rowCount() == 1 ) {
20             // Get values
21             $first = $row[ 'first_name' ];
22             $last  = $row[ 'last_name' ];
23
24             // Feedback for end user
25             echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}</pre>";
26         }
27     }
28 }
29
30 // Generate Anti-CSRF token
31 generateSessionToken();
32
33 ?>
```


Prepared Statements in NodeJS

```
1 var name = "Bumblebee";
2 var lastname = "Bees";
3 var Tel = "0642222222";
4
5 sql_stmt = "insert into emp values(?,?,?)";
6
7 var values = [name, lastname, Tel];
8
9 sql_stmt = mysql.format(sql_stmt, values);
10
11 connection.query(sql_stmt, function (error, result) {
12     if (error) {
13         console.log(error.message);
14     }
15     console.log(result.insertId);
16 });
```



Input Validation

- ☐ Overview
- ☐ SQL Injection
- ☒ **Command Injection**
- ☐ Directory Traversal
- ☐ Blacklist vs. Whitelist validation
- ☐ Regular expressions(Regex)
- ☐ Client Side vs Server Side Validation

Failed Code #1

```
1 <?php
2
3 if( isset( $_POST[ 'Submit' ] ) ) {
4     // Get input
5     $target = $_REQUEST[ 'ip' ];
6
7     // Determine OS and execute the ping command.
8     if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
9         // Windows
10         $cmd = shell_exec( 'ping ' . $target );
11     }
12     else {
13         // *nix
14         $cmd = shell_exec( 'ping -c 4 ' . $target );
15     }
16
17     // Feedback for the end user
18     echo "<pre>{$cmd}</pre>";
19 }
20
21 ?>
```

Ping a device

Enter an IP address:

Submit

```
PING www.google.com (172.217.166.132): 56 data bytes
64 bytes from 172.217.166.132: icmp_seq=0 ttl=53 time=41.703 ms
64 bytes from 172.217.166.132: icmp_seq=1 ttl=53 time=62.646 ms
64 bytes from 172.217.166.132: icmp_seq=2 ttl=53 time=46.371 ms
64 bytes from 172.217.166.132: icmp_seq=3 ttl=53 time=40.103 ms
--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 40.103/47.706/62.646/8.928 ms
```

Ping a device

Enter an IP address:

www.google.com

Submit

Command Injection

Ping a device

Enter an IP address:

```
PING www.google.com (172.217.26.132): 56 data bytes
64 bytes from 172.217.26.132: icmp_seq=0 ttl=53 time=40.254 ms
64 bytes from 172.217.26.132: icmp_seq=1 ttl=53 time=39.042 ms
64 bytes from 172.217.26.132: icmp_seq=2 ttl=53 time=42.334 ms
64 bytes from 172.217.26.132: icmp_seq=3 ttl=53 time=40.472 ms
--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 39.042/40.526/42.334/1.178 ms
.
..
help
index.php
source
```

Ping a device

Enter an IP address:

```
PING www.google.com (172.217.24.164): 56 data bytes
64 bytes from 172.217.24.164: icmp_seq=0 ttl=53 time=39.735 ms
64 bytes from 172.217.24.164: icmp_seq=1 ttl=53 time=39.317 ms
64 bytes from 172.217.24.164: icmp_seq=2 ttl=53 time=39.564 ms
64 bytes from 172.217.24.164: icmp_seq=3 ttl=53 time=51.646 ms
--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 39.317/42.566/51.646/5.245 ms
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
_apt:x:100:65534:/nonexistent:/bin/false
mysql:x:101:101:MySQL Server,,:/nonexistent:/bin/false
```

Failed Code #2

```
1  <?php
2
3  if( isset( $_POST[ 'Submit' ] ) ) {
4      // Get input
5      $target = $_REQUEST[ 'ip' ];
6
7      // Set blacklist
8      $substitutions = array(
9          '&&' => '',
10         ';'  => '',
11     );
12
13     // Remove any of the characters in the array (blacklist).
14     $target = str_replace( array_keys( $substitutions ), $substitutions, $target );
15
16     // Determine OS and execute the ping command.
17     if( striistr( php_uname( 's' ), 'Windows NT' ) ) {
18         // Windows
19         $cmd = shell_exec( 'ping ' . $target );
20     }
21     else {
22         // *nix
23         $cmd = shell_exec( 'ping -c 4 ' . $target );
24     }
25
26     // Feedback for the end user
27     echo "<pre>{$cmd}</pre>";
28 }
29
30 ?>
```




Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

Vulnerability: File Inclusion

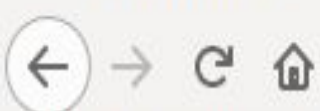
File 1

Hello **admin**
Your IP address is: **192.168.1.100**

[\[back\]](#)

More info

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3



```
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 systemd-network:x:100:102:systemd Network Management,,:/run
/systemd/netif:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd/resolve:/usr/sbin/nologin syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin _apt:x:104:65534::/nonexistent:/usr/sbin/nologin uidd:x:105:111::/run/uidd:/usr/sbin/nologin avahi-autoipd:x:106:112:Avahi autoip
daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin usbmux:x:107:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin dnsmasq:x:108:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,:/proc:/usr/sbin/nologin cups-pk-helper:x:110:116:user for cups-pk-helper service,,:/home/cups-pk-helper:/usr/sbin/nologin speech-dispatcher:x:111:29:Speech
Dispatcher,,:/var/run/speech-dispatcher:/bin/false whoopsie:x:112:117::/nonexistent:/bin/false kernoops:x:113:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin saned:x:114:119:/var
/lib/saned:/usr/sbin/nologin pulse:x:115:120:PulseAudio daemon,,:/var/run/pulse:/usr/sbin/nologin avahi:x:116:122:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin hplip:x:118:7:HPLIP system user,,:/var/run/hplip:/bin/false geoclue:x:119:124:/var/lib/geoclue:/usr/sbin
/nologin gnome-initial-setup:x:120:65534:/run/gnome-initial-setup:/bin/false gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false ubuntu:x:1000:1000:ubuntu,,:/home/ubuntu:
/bin/bash mysql:x:122:127:MySQL Server,,:/nonexistent:/bin/false
```

[Home](#)[Instructions](#)[Setup / Reset DB](#)[Brute Force](#)



```
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
20 systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
21 syslog:x:102:106::/home/syslog:/usr/sbin/nologin
22 messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
23 _apt:x:104:65534::/nonexistent:/usr/sbin/nologin
24 uidd:x:105:111::/run/uidd:/usr/sbin/nologin
25 avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
26 usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
27 dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
28 rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
29 cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
30 speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
31 whoopsie:x:112:117::/nonexistent:/bin/false
32 kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
33 saned:x:114:119::/var/lib/saned:/usr/sbin/nologin
34 pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
35 avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
36 colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
37 hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
38 geoclue:x:119:124::/var/lib/geoclue:/usr/sbin/nologin
39 gnome-initial-setup:x:120:65534::/run/gnome-initial-setup:/bin/false
40 gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
41 ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
42 mysql:x:122:127:MySQL Server,,,:/nonexistent:/bin/false
43
```

Failed Code #1

```
1  <?php
2
3  // The page we wish to display
4  $file = $_GET[ 'page' ];
5
6  ?>
```


Failed Code #2

```
1  <?php
2
3  // The page we wish to display
4  $file = $_GET[ 'page' ];
5
6  // Input validation
7  $file = str_replace( array( "http://", "https://" ), "", $file );
8  $file = str_replace( array( "../", "..\\" ), "", $file );
9
10 ?>
```

Should we implement input validation on

Client Side or Server Side?

Input Validation On Both Side



Server side

- ❖ Server is trusted system

Client side

- ❖ Reduce bad requests
- ❖ Reduce server processing
- ❖ User friendly interface

Input Validation Check list

- ☐ Conduct all data validation on a trusted system (e.g., The server)
- ☐ Identify all data sources and classify them into trusted and untrusted. Validate all data from untrusted sources (e.g., Databases, file streams, etc.)
- ☐ There should be a centralized input validation routine for the application
- ☐ Specify proper character sets, such as UTF-8, for all sources of input
- ☐ Encode data to a common character set before validating (Canonicalize)
- ☐ All validation failures should result in input rejection
- ☐ Determine if the system supports UTF-8 extended character sets and if so, validate after UTF-8 decoding is completed
- ☐ Validate all client provided data before processing, including all parameters, URLs and HTTP header content (e.g. Cookie names and values). Be sure to include automated post backs from JavaScript, Flash or other embedded code
- ☐ Verify that header values in both requests and responses contain only ASCII characters
- ☐ Validate data from redirects (An attacker may submit malicious content directly to the target of the redirect, thus circumventing application logic and any validation performed before the redirect)
- ☐ Validate for expected data types
- ☐ Validate data range
- ☐ Validate data length
- ☐ Validate all input against a "white" list of allowed characters, whenever possible
- ☐ If any potentially hazardous characters must be allowed as input, be sure that you implement additional controls like output encoding, secure task specific APIs and accounting for the utilization of that data throughout the application. Examples of common hazardous characters include: < > " ' % () & + \\'\"
- ☐ If your standard validation routine cannot address the following inputs, then they should be checked discretely
 - ☐ Check for null bytes (%00)
 - ☐ Check for new line characters (%0d, %0a, \r, \n)
 - ☐ Check for "dot-dot-slash" (../ or ..\) path alterations characters. In cases where UTF-8 extended character set encoding is supported, address alternate representation like: %c0%ae%c0%ae/
 - ☐ (Utilize canonicalization to address double encoding or other forms of obfuscation attacks)

A2: Broken Authentication and Session Management



Strong Login System



What is authentication

- ❑ **Identification** : who someone is (e.g. username, Smart Card, ID Card)
- ❑ **Authentication** : The process of proving an identity (e.g. password)
- ❑ **Authorization** : What are you allowed to do? (e.g. Read/Write access)
- ❑ **Verification** : Confirming the truth or accuracy of the details provided by user

Authentication and Password Management

Authentication is the first step in access control, and there are three common factors used for authentication

something you know	: Username & Password
something you have	: Credit Card & ATM Card
something you are	: Fingerprint & biometric method

Users enumeration (Failed)

```
// Validate credentials
if(empty($username_err) && empty($password_err)){
    // Prepare a select statement
    $sql = "SELECT id, username, password FROM users WHERE username = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_username);

        // Set parameters
        $param_username = $username;

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            // Store result
            mysqli_stmt_store_result($stmt);

            // Check if username exists, if yes then verify password
            if(mysqli_stmt_num_rows($stmt) == 1){
                // Bind result variables
                mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);
                if(mysqli_stmt_fetch($stmt)){
                    if(password_verify($password, $hashed_password)){
                        // Password is correct, so start a new session
                        session_start();

                        // Store data in session variables
                        $_SESSION["loggedin"] = true;
                        $_SESSION["id"] = $id;
                        $_SESSION["username"] = $username;

                        // Redirect user to welcome page
                        header("location: welcome.php");
                    } else{
                        // Display an error message if password is not valid
                        $password_err = "The password you entered was not valid.";
                    }
                }
            } else{
                // Display an error message if username doesn't exist
                $username_err = "No account found with that username.";
            }
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
```

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

No account found with that username.

The password you entered was not valid.

Users enumeration (Secure)

```
// Validate credentials
if(empty($username) && empty($password)){
    // Prepare a select statement
    $sql = "SELECT id, username, password FROM users WHERE username = ? and password = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $username);
        mysqli_stmt_bind_param($stmt, "s", hash($password));

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            // Store result
            mysqli_stmt_store_result($stmt);

            // Check if username exists, if yes then verify password
            if(mysqli_stmt_num_rows($stmt) == 1){
                // Bind result variables
                mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);
                if(mysqli_stmt_fetch($stmt)){
                    session_start();

                    // Store data in session variables
                    $_SESSION["loggedin"] = true;
                    $_SESSION["id"] = $id;
                    $_SESSION["username"] = $username;

                    // Redirect user to welcome page
                    header("location: welcome.php");
                }
            } else{
                // Display an error message if username doesn't exist
                $username_err = "Username or Password is invalid";
            }
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }
    }

    // Close statement
    mysqli_stmt_close($stmt);
}
```

Login

Please fill in your credentials to login.

Username

Password

Login

Don't have an account? [Sign up now.](#)

Username or Password is invalid

Login Function Security

- ☐ When user login failed, do not tell user the failed reason. Response the same message.
- ☐ Lock user's account when user login failed for many times
- ☐ Delay User login failed

Password Attack

- Do not allow user set easy password such as
 - Well known
 - Dictionary
 - Simple string

Anti-Automation

- Check user agent such as
 - hydra

Authentication & Password Management Check List

- ☐ Require authentication for all pages and resources, except those specifically intended to be public
- ☐ All authentication controls must be enforced on a trusted system (e.g., The server)
- ☐ Establish and utilize standard, tested, authentication services whenever possible
- ☐ Use a centralized implementation for all authentication controls, including libraries that call external authentication services
- ☐ Segregate authentication logic from the resource being requested and use redirection to and from the centralized authentication control
- ☐ All authentication controls should fail securely
- ☐ All administrative and account management functions must be at least as secure as the primary authentication mechanism
- ☐ If your application manages a credential store, it should ensure that only cryptographically strong one-way salted hashes of passwords are stored and that the table/file that stores the passwords and keys is write-able only by the application. (Do not use the MD5 algorithm if it can be avoided)
- ☐ Password hashing must be implemented on a trusted system (e.g., The server).
- ☐ Validate the authentication data only on completion of all data input, especially for sequential authentication implementations
- ☐ Authentication failure responses should not indicate which part of the authentication data was incorrect. For example, instead of "Invalid username" or "Invalid password", just use "Invalid username and/or password" for both. Error responses must be truly identical in both display and source code
- ☐ Utilize authentication for connections to external systems that involve sensitive information or functions
- ☐ Authentication credentials for accessing services external to the application should be encrypted and stored in a protected location on a trusted system (e.g., The server). The source code is NOT a secure location
- ☐ Use only HTTP POST requests to transmit authentication credentials
- ☐ Only send non-temporary passwords over an encrypted connection or as encrypted data, such as in an encrypted email. Temporary passwords associated with email resets may be an exception
- ☐ Enforce password complexity requirements established by policy or regulation. Authentication credentials should be sufficient to withstand attacks that are typical of the threats in the deployed environment. (e.g., requiring the use of alphabetic as well as numeric and/or special characters)

Authentication & Password Management Check List

- ☐ Enforce password length requirements established by policy or regulation. Eight characters is commonly used, but 16 is better or consider the use of multi-word pass phrases
- ☐ Password entry should be obscured on the user's screen. (e.g., on web forms use the input type "password")
- ☐ Enforce account disabling after an established number of invalid login attempts (e.g., five attempts is common). The account must be disabled for a period of time sufficient to discourage brute force guessing of credentials, but not so long as to allow for a denial-of-service attack to be performed
- ☐ Password reset and changing operations require the same level of controls as account creation and authentication.
- ☐ Password reset questions should support sufficiently random answers. (e.g., "favourite book" is a bad question because "The Bible" is a very common answer)
- ☐ If using email based resets, only send email to a pre-registered address with a temporary link/password
- ☐ Temporary passwords and links should have a short expiration time
- ☐ Enforce the changing of temporary passwords on the next use
- ☐ Notify users when a password reset occurs
- ☐ Prevent password re-use
- ☐ Passwords should be at least one day old before they can be changed, to prevent attacks on password re-use
- ☐ Enforce password changes based on requirements established in policy or regulation. Critical systems may require more frequent changes. The time between resets must be administratively controlled
- ☐ Disable "remember me" functionality for password fields
- ☐ The last use (successful or unsuccessful) of a user account should be reported to the user at their next successful login
- ☐ Implement monitoring to identify attacks against multiple user accounts, utilizing the same password. This attack pattern is used to bypass standard lockouts, when user IDs can be harvested or guessed
- ☐ Change all vendor-supplied default passwords and user IDs or disable the associated accounts
- ☐ Re-authenticate users prior to performing critical operations
- ☐ Use Multi-Factor Authentication for highly sensitive or high value transactional accounts
- ☐ If using third party code for authentication, inspect the code carefully to ensure it is not affected by any malicious code

A3: Sensitive Data Exposure

What is Sensitive Data?

Information such as:

Bank account details

Credit card numbers

Passwords

Session tokens

Tax details

Company secrets

Healthcare information

Contact and demographic information

amongst others can be considered to be Sensitive Data.

A4: XML External Entities (XXE)

A4: XML External Entities (XXE)

XML external entity injection (also known as XXE) is a web security vulnerability that allows an attacker to interfere with an application's processing of XML data. It often allows an attacker to view files on the application server filesystem, and to interact with any back-end or external systems that the application itself can access.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

```
<!ENTITY xxe SYSTEM "https://192.168.1.1/private" >]>
```

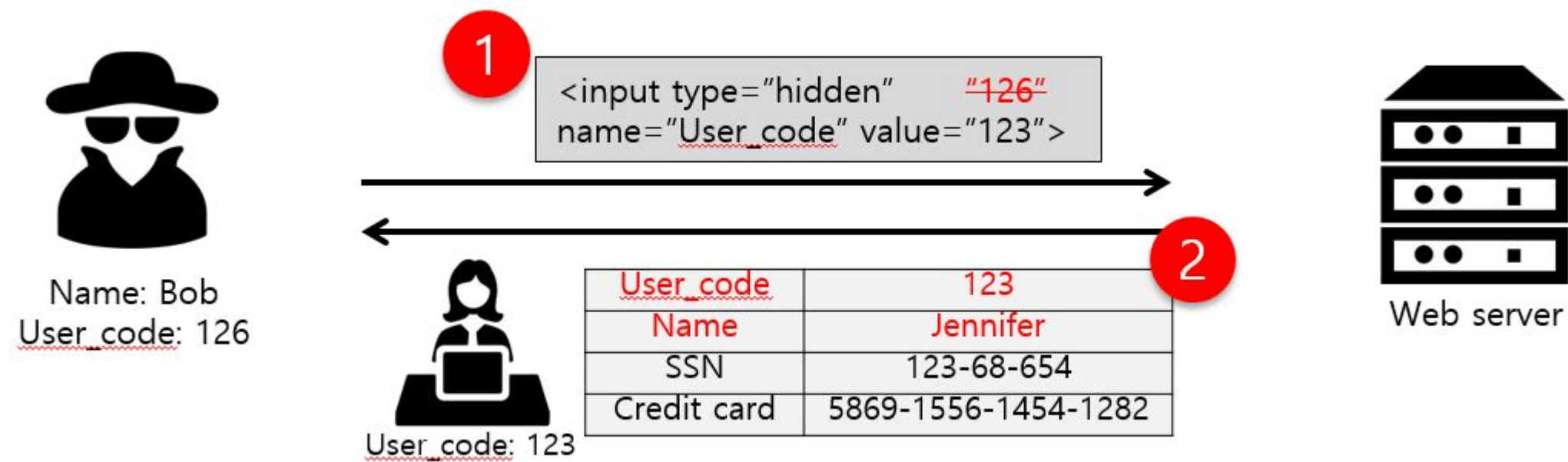
A5: Broken Access Control

A5: Broken Access Control

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification or destruction of all data, or performing a business function outside of the limits of the user. Common access control vulnerabilities include:

- * Bypassing access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool.
- * Allowing the primary key to be changed to another's users record, permitting viewing or editing someone else's account.
- * Elevation of privilege. Acting as a user without being logged in, or acting as an admin when logged in as a user.
- * Metadata manipulation, such as replaying or tampering with a JSON Web Token (JWT) access control token or a cookie or hidden field manipulated to elevate privileges, or abusing JWT invalidation.
- * CORS misconfiguration allows unauthorized API access.
- * Force browsing to authenticated pages as an unauthenticated user or to privileged pages as a standard user. Accessing API with missing access controls for POST, PUT and DELETE.

A5: Broken Access Control



A6: Security Misconfiguration

A6: Security Misconfiguration

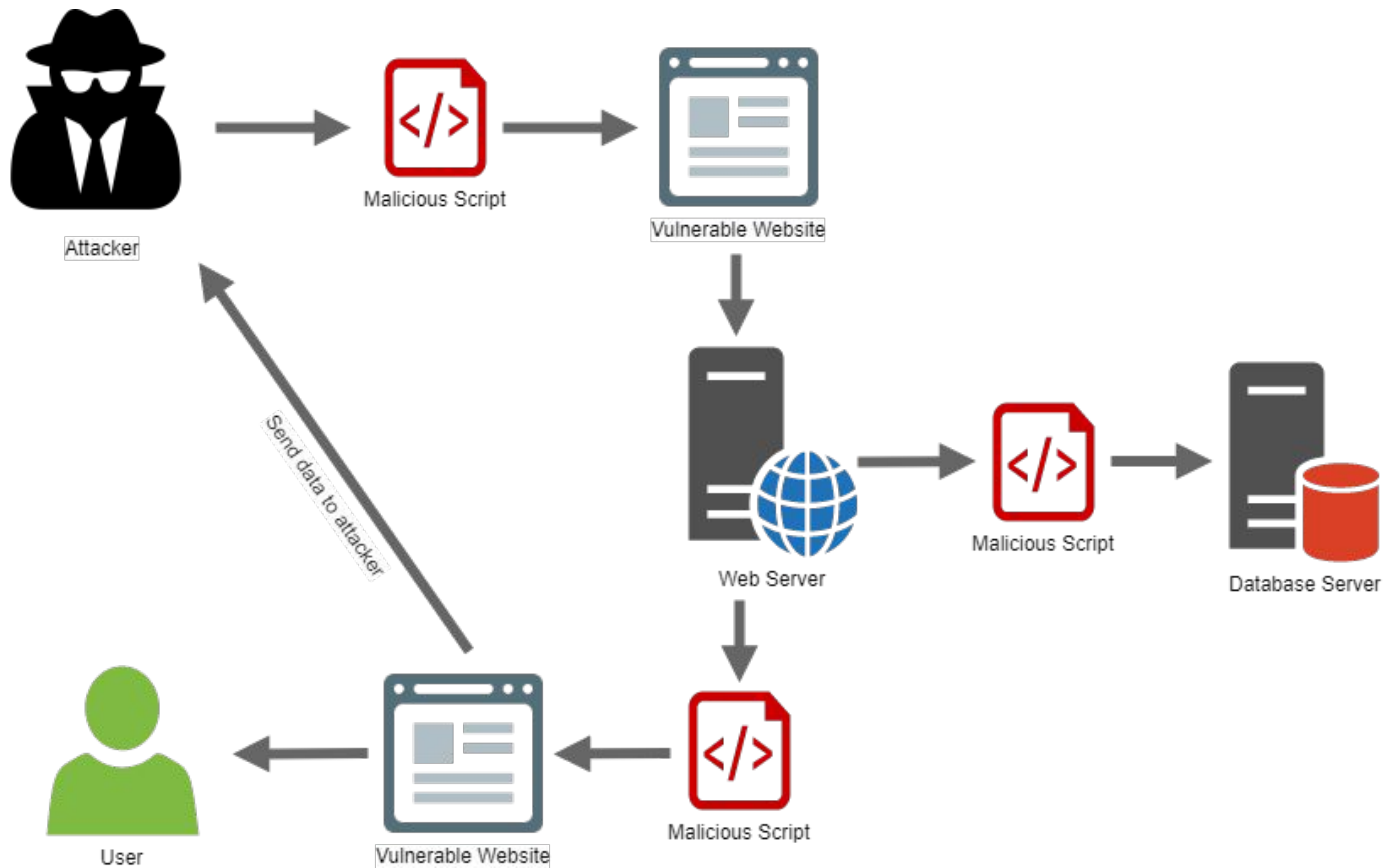
1. directory listing enable
2. update security patch out date
3. HttpOnly Cookie not implement
4. Error message
5.

Index of /data

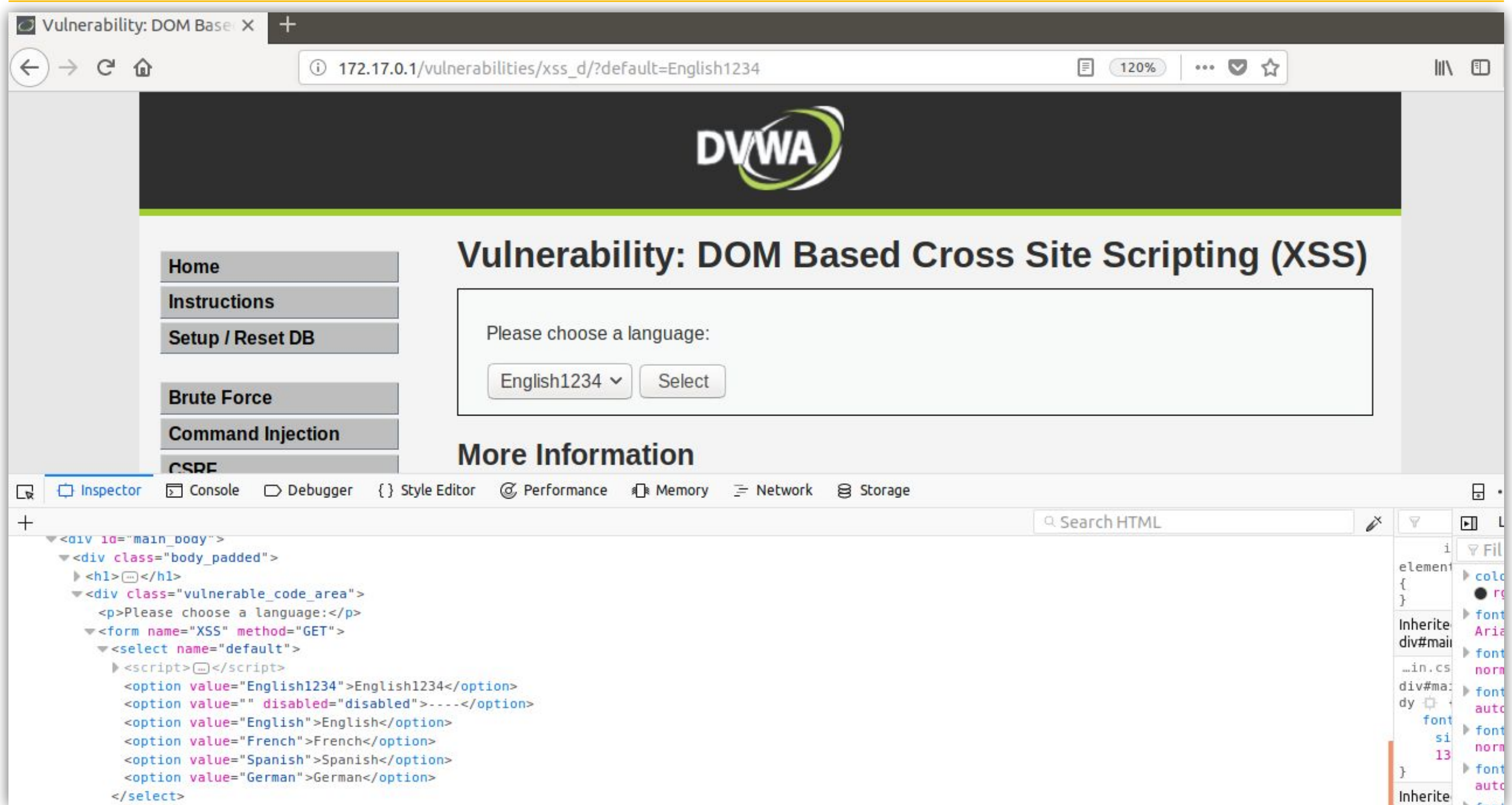
<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 01.mp4	2012-12-11 07:52	25M	
 02.mp4	2012-12-11 08:05	34M	
 get.php	2014-09-09 01:01	29	
 post.php	2014-09-09 01:03	139	

Apache/2.4.7 (Ubuntu) Server at www.wised.com Port 443

A7: Cross-Site Scripting (XSS)



Cross Site Scripting (Dom Based XSS)



Vulnerability: DOM Base X

172.17.0.1/vulnerabilities/xss_d/?default=English1234

120%

DVWA

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English1234 ▼ Select

More Information

Inspector Console Debugger Style Editor Performance Memory Network Storage

Search HTML

```
<div id="main_body">
  <div class="body_padded">
    <h1></h1>
    <div class="vulnerable_code_area">
      <p>Please choose a language:</p>
      <form name="XSS" method="GET">
        <select name="default">
          <script></script>
          <option value="English1234">English1234</option>
          <option value="" disabled="disabled">-----</option>
          <option value="English">English</option>
          <option value="French">French</option>
          <option value="Spanish">Spanish</option>
          <option value="German">German</option>
        </select>
      </form>
    </div>
  </div>
</div>
```

Failed Code #1

```
1 <?php
2
3 header ("X-XSS-Protection: 0");
4
5 // Is there any input?
6 if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
7     // Feedback for end user
8     echo '<pre>Hello ' . $_GET[ 'name' ] . '</pre>';
9 }
10
11 ?>
```

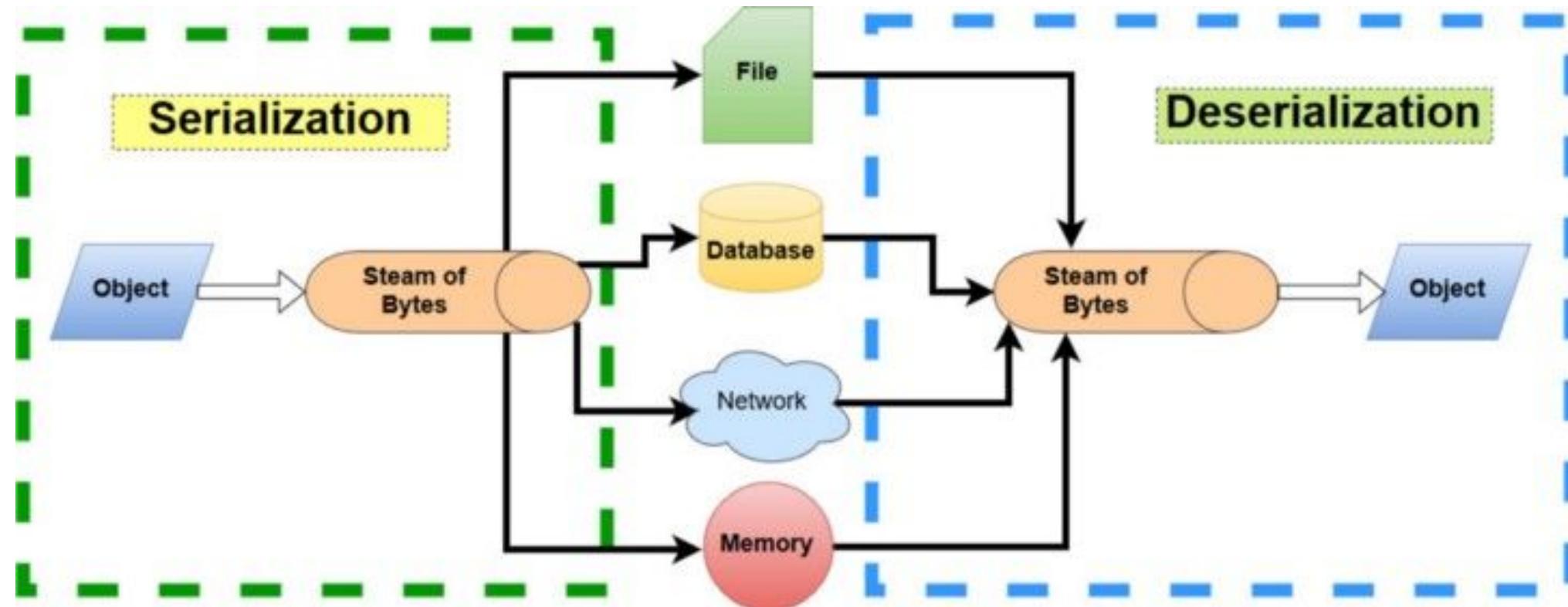
Secure Code

```
1 <?php
2
3 // Is there any input?
4 if( array_key_exists( "name", $_GET ) && $_GET[ 'name' ] != NULL ) {
5     // Check Anti-CSRF token
6     checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );
7
8     // Get input
9     $name = htmlspecialchars( $_GET[ 'name' ] );
10
11     // Feedback for end user
12     echo "<pre>Hello ${name}</pre>";
13 }
14
15 // Generate Anti-CSRF token
16 generateSessionToken();
17
18 ?>
```


A8: Insecure Deserialization

A8: Insecure Deserialization

Exploitation of deserialization is somewhat difficult, as off the shelf exploits rarely work without changes or tweaks to the underlying exploit code.



YOU SHALL NOT PASS



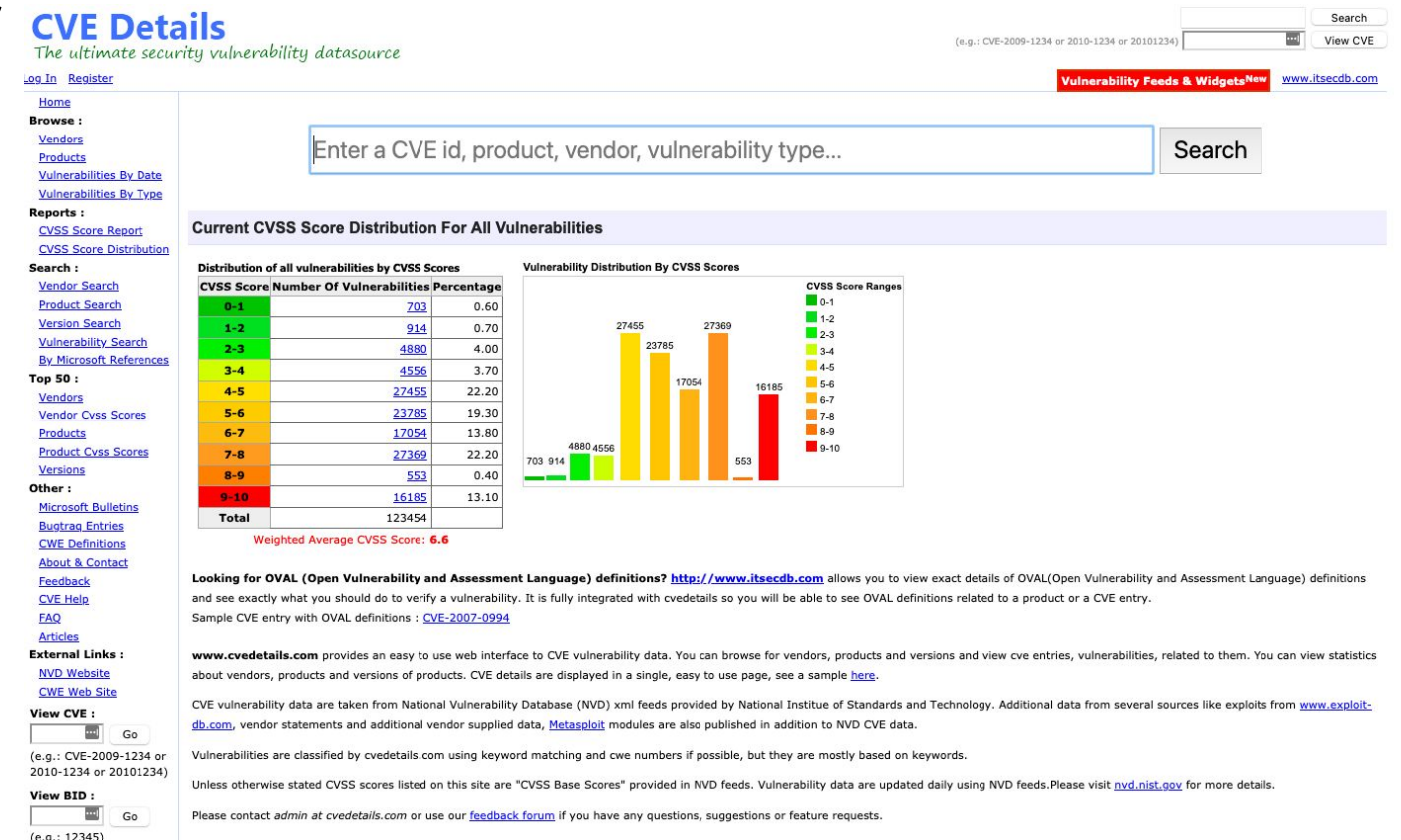
USER-CONTENT TO unserialize()

A9: Using Known Vulnerable Components

A9: Using Known Vulnerable Components

While it is easy to find already-written exploits for many known vulnerabilities, other vulnerabilities require concentrated effort to develop a custom exploit.

1. Update Library , framework version, runtime, software, OS.
2. Check CVE Update from (<https://cve.mitre.org>), NVD (<https://nvd.nist.gov/>)
3. Using only well-know Library , Framework, Plugin
4. Update Library , Framework, Plugin to the latest version.



A10: Insufficient Logging & Monitoring

A10: Insufficient Logging & Monitoring

Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident.

Attackers rely on the lack of monitoring and timely response to achieve their goals without being detected.

Centralized Logging



<https://www.digitalocean.com/community/tutorials/building-for-production-web-applications-centralized-logging>

Access Control Check List

- ☐ Use only trusted system objects, e.g. server side session objects, for making access authorization decisions
- ☐ Use a single site-wide component to check access authorization. This includes libraries that call external authorization services
- ☐ Access controls should fail securely
- ☐ Deny all access if the application cannot access its security configuration information
- ☐ Enforce authorization controls on every request, including those made by server side scripts, "includes" and requests from rich client-side technologies like AJAX and Flash
- ☐ Segregate privileged logic from other application code
- ☐ Restrict access to files or other resources, including those outside the application's direct control, to only authorized users
- ☐ Restrict access to protected URLs to only authorized users
- ☐ Restrict access to protected functions to only authorized users
- ☐ Restrict direct object references to only authorized users
- ☐ Restrict access to services to only authorized users
- ☐ Restrict access to application data to only authorized users
- ☐ Restrict access to user and data attributes and policy information used by access controls
- ☐ Restrict access security-relevant configuration information to only authorized users
- ☐ Server side implementation and presentation layer representations of access control rules must match
- ☐ If state data must be stored on the client, use encryption and integrity checking on the server side to catch state tampering.
- ☐ Enforce application logic flows to comply with business rules
- ☐ Limit the number of transactions a single user or device can perform in a given period of time. The transactions/time should be above the actual business requirement, but low enough to deter automated attacks
- ☐ Use the "referer" header as a supplemental check only, it should never be the sole authorization check, as it is can be spoofed
- ☐ If long authenticated sessions are allowed, periodically re-validate a user's authorization to ensure that their privileges have not changed and if they have, log the user out and force them to re-authenticate
- ☐ Implement account auditing and enforce the disabling of unused accounts (e.g., After no more than 30 days from the expiration of an account's password.)
- ☐ The application must support disabling of accounts and terminating sessions when authorization ceases (e.g., Changes to role, employment status, business process, etc.)
- ☐ Service accounts or accounts supporting connections to or from external systems should have the least privilege possible
- ☐ Create an Access Control Policy to document an application's business rules, data types and access authorization criteria and/or processes so that access can be properly provisioned and controlled. This includes identifying access requirements for both the data and system resources

Database Security

- ☐ Use strongly typed parameterized queries
- ☐ Utilize input validation and output encoding and be sure to address meta characters. If these fail, do not run the database command
- ☐ Ensure that variables are strongly typed
- ☐ The application should use the lowest possible level of privilege when accessing the database
- ☐ Use secure credentials for database access
- ☐ Connection strings should not be hard coded within the application. Connection strings should be stored in a separate configuration file on a trusted system and they should be encrypted.
- ☐ Use stored procedures to abstract data access and allow for the removal of permissions to the base tables in the database
- ☐ Close the connection as soon as possible
- ☐ Remove or change all default database administrative passwords. Utilize strong passwords/phrases or implement multi-factor authentication
- ☐ Turn off all unnecessary database functionality (e.g., unnecessary stored procedures or services, utility packages, install only the minimum set of features and options required (surface area reduction))
- ☐ Remove unnecessary default vendor content (e.g., sample schemas)
- ☐ Disable any default accounts that are not required to support business requirements
- ☐ The application should connect to the database with different credentials for every trust distinction (e.g., user, read-only user, guest, administrators)

File Upload

- Filenames threats
- File extension handling
- Null-byte injection

File Management

- ☐ Do not pass user supplied data directly to any dynamic include function
- ☐ Require authentication before allowing a file to be uploaded
- ☐ Limit the type of files that can be uploaded to only those types that are needed for business purposes
- ☐ Validate uploaded files are the expected type by checking file headers. Checking for file type by extension alone is not sufficient
- ☐ Do not save files in the same web context as the application. Files should either go to the content server or in the database.
- ☐ Prevent or restrict the uploading of any file that may be interpreted by the web server.
- ☐ Turn off execution privileges on file upload directories
- ☐ Implement safe uploading in UNIX by mounting the targeted file directory as a logical drive using the associated path or the chrooted environment
- ☐ When referencing existing files, use a white list of allowed file names and types. Validate the value of the parameter being passed and if it does not match one of the expected values, either reject it or use a hard coded default file value for the content instead
- ☐ Do not pass user supplied data into a dynamic redirect. If this must be allowed, then the redirect should accept only validated, relative path URLs
- ☐ Do not pass directory or file paths, use index values mapped to pre-defined list of paths
- ☐ Never send the absolute file path to the client
- ☐ Ensure application files and resources are read-only
- ☐ Scan user uploaded files for viruses and malware

Ref:

<https://owasp.org/www-project-top-ten/2017/>



Join! us!