

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**AN TOÀN ỨNG DỤNG WEB  
& CƠ SỞ DỮ LIỆU**

**Đề tài:**

**XY DỰNG WEBSITE QUẢN LÝ ĐIỂM SINH VIÊN  
CÓ SỬ DỤNG BIỆN PHÁP BẢO MẬT WEB &  
CƠ SỞ DỮ LIỆU**

**Giảng viên hướng dẫn: PHAN NGHĨA HIỆP**

**Nhóm thực hiện: Nhóm 10**

<b>Mã số sinh viên:</b>	<b>TRẦN ANH DŨNG</b>	<b>N19DCAT013</b>
	<b>NGUYỄN QUANG CHÍNH</b>	<b>N19DCAT011</b>
	<b>MAI THẾ CHUYỀN EM</b>	<b>N19DCAT018</b>
	<b>NGUYỄN HOÀNG ĐẠI NGHĨA</b>	<b>N19DCAT054</b>
	<b>TRẦN THANH TRÀ</b>	<b>N19DCAT089</b>

**Lớp: D19CQAT01-N**

**TP. HCM, 12/2022**



**AN TOÀN ỨNG DỤNG WEB  
& CƠ SỞ DỮ LIỆU**

*Đề tài:*

**XY DỰNG WEBSITE QUẢN LÝ ĐIỂM SINH VIÊN  
CÓ SỬ DỤNG BIỆN PHÁP BẢO MẬT WEB &  
CƠ SỞ DỮ LIỆU**

**Giảng viên hướng dẫn: PHAN NGHĨA HIỆP**

**Nhóm thực hiện: Nhóm 10**

<b>Mã số sinh viên:</b>	<b>TRẦN ANH DŨNG</b>	<b>N19DCAT013</b>
	<b>NGUYỄN QUANG CHÍNH</b>	<b>N19DCAT011</b>
	<b>MAI THẾ CHUYỀN EM</b>	<b>N19DCAT018</b>
	<b>NGUYỄN HOÀNG ĐẠI NGHĨA</b>	<b>N19DCAT054</b>
	<b>TRẦN THANH TRÀ</b>	<b>N19DCAT089</b>

**Lớp: D19CQAT01-N**

**TP. HCM, 12/2022**

## NHẬN XÉT CỦA GVHD

[illegible]

## LỜI CẢM ƠN

Đề tài “Xây dựng Website Quản Lý Điểm Sinh Viên có sử dụng các biện pháp bảo mật Web & Cơ sở dữ liệu” là đề tài Nhóm 10 lựa chọn để nghiên cứu và làm đồ án báo cáo học phần “An Toàn Ứng Dụng Web & Cơ Sở Dữ Liệu” thuộc chương trình đại học ngành An Toàn Thông Tin tại Học Viện Công Nghệ Bưu Chính Viễn Thông – Cơ Sở Tại Thành phố Hồ Chí Minh.

Để hoàn thành quá trình nghiên cứu và hoàn thiện đồ án học phần này, lời đầu tiên Nhóm 10 xin gửi lời cảm ơn chân thành tới Quý Thầy, Cô, bạn bè của Học Viện Công Nghệ Bưu Chính Viễn Thông – Cơ Sở Tại Thành phố Hồ Chí Minh.

Bày tỏ lòng biết ơn sâu sắc nhất đến thầy, cô trong khoa Công Nghệ Thông Tin 2 đã dìu dắt, chia sẻ những kiến thức quý báu trong suốt quá trình học tập tại trường. Đặc biệt là Thầy ThS. Phan Nghĩa Hiệp cùng với tri thức và tâm huyết của Thầy đã tạo điều kiện cho Nhóm 10 hoàn thành đồ án.

Cuối cùng, Nhóm 10 xin cảm ơn những người thân, bạn bè luôn động viên, sẻ chia, giúp đỡ, cổ vũ tinh thần... Đó là nguồn động lực giúp Nhóm 10 hoàn thành tốt đồ án học phần này.

Nhóm xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày ... tháng ... năm ...

Đại diện Sinh viên

Trần Anh Dũng

## MỤC LỤC

NHẬN XÉT CỦA GVHD .....	3
LỜI CẢM ƠN.....	4
MỤC LỤC .....	5
DANH MỤC CÁC HÌNH.....	8
NỘI DUNG BÁO CÁO .....	10
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....	10
1.1. Sơ lược ứng dụng:.....	10
1.2. Các đối tượng cần bảo mật trong ứng dụng: .....	10
1.2.1. Account.....	10
1.2.2. Cơ sở dữ liệu (Database): .....	11
CHƯƠNG 2: THỰC NGHIỆM.....	11
2.1. An Toàn Ứng Dụng Web.....	11
2.1.1. Chèn mã HTML và Cross-site Scripting (XSS).....	11
2.1.1.1. Vị trí có thể bị chèn mã XSS .....	11
2.1.1.2. Phòng chống .....	11
2.1.1.3. Thử nghiệm.....	12
2.1.2. SQL injection.....	12
2.1.2.1. Vị trí có thể chèn mã SQL injection.....	12
2.1.2.2. Phòng chống .....	12
2.1.2.3. Thử nghiệm.....	13
2.1.3. Authentication Attack.....	14
2.1.3.1. Vị trí có thể bị lỗi Authentication.....	14
2.1.3.2. Phòng chống .....	14

2.1.4. Logic Attack .....	15
2.1.5. Tấn công vào trình duyệt web và sự riêng tư của người dùng 16	
2.1.6. Xác thực người dùng và trao quyền truy cập (Access control) 16	
2.1.7. Bảo mật phiên làm việc .....	18
2.1.8. Bảo vệ máy chủ web.....	19
2.1.9. Bảo mật hệ thống file.....	19
2.2. An Toàn Cơ Sở Dữ Liệu.....	20
2.2.1. Default and Weak Passwords .....	20
2.2.1.1. Mô tả.....	20
2.2.1.2. Phòng chống .....	20
2.2.1.3. Thử nghiệm.....	20
2.2.2. SQL Injection in the DBMS ( Lỗi chèn mã SQL).....	21
2.2.2.1. Mô tả.....	21
2.2.2.2. Phòng chống .....	22
2.2.2.3. Thử nghiệm.....	22
2.2.3. Excessive User & Group Privileges .....	22
2.2.3.1. Mô tả.....	22
2.2.3.2. Phòng chống .....	22
2.2.3.3. Thử nghiệm.....	23
2.2.4. Unnecessary Enabled DBMS Features.....	25
2.2.4.1. Mô tả.....	25
2.2.4.2. Phòng chống .....	26

2.2.4.3. Thử nghiệm.....	26
2.2.5. Unpathed Database .....	27
2.2.5.1. Mô tả.....	27
2.2.5.2. Phòng chống .....	28
2.2.5.3. Thử nghiệm.....	28
2.2.6. Unencrypted Data .....	29
2.2.6.1. Mô tả.....	29
2.2.6.2. Phòng chống .....	29
2.2.6.3. Thử nghiệm.....	29
TÀI LIỆU THAM KHẢO .....	31
BẢNG PHÂN CÔNG CÔNG VIỆC .....	32

## DANH MỤC CÁC HÌNH

Figure 1: Thử nghiệm XSS .....	12
Figure 2: Thực hiện SQLi để bypass login .....	13
Figure 3: Kết quả trả về của PreparedStatement là 0 (null).....	13
Figure 4: Mã hoá mật khẩu lưu trong cơ sở dữ liệu .....	13
Figure 5: Mã hoá RSA trên Server .....	13
Figure 6: Thử nghiệm Captcha .....	15
Figure 7: Xử lý request GET/POST với endpoint sinh-vien .....	15
Figure 8: Người dùng phải đăng nhập trước đó để có session truy cập vào các endpoint .....	16
Figure 9: HTML entity các giá trị.....	16
Figure 10: Authentication .....	17
Figure 11: Người dùng phải login thì mới có thể có cooke để truy cập vào các endpoint .....	17
Figure 12: Nhân viên thì sẽ không thể thêm nhân viên mới.....	17
Figure 13: Admin có thêm nhân viên mới .....	18
Figure 14: Bảo mật phiên làm việc.....	18
Figure 15: Tính năng Log-Out.....	19
Figure 16: Bảo mật máy chủ Web .....	19
Figure 17: Thay đổi mật khẩu mạnh cho tài khoản mặc định SA .....	21
Figure 18: Kết nối trong ứng dụng web.....	21
Figure 19: Sử dụng thư viện để thực hiện câu truy vấn chống SQLi .....	22
Figure 20: Tạo User DBRead chỉ có quyền trên database QLDSV .....	23
Figure 21: User chỉ có quyền đọc trong Database QLDSV .....	24
Figure 22: Thử trường hợp User DBRead muốn quyền Tạo Database .....	25
Figure 23: Thực hiện câu lệnh "dir *exe" xuất danh sách file trên SQL Server .....	26



Figure 24: Tắt cấu hình cho phép "Show Advanced Option" để tắt xp_cmdshell .....	27
Figure 25: Không thực hiện OS Command sau khi tắt xp_cmdshell .....	27
Figure 26: Danh sách các CVE của SQL Server 2000 .....	28
Figure 27: Danh sách các CVE của SQL Server Studio 18.....	28
Figure 28: SP để thêm sinh viên có mật khẩu đã được mã hóa.....	29
Figure 29: Thực hiện gọi đến SP_INS_SINHVIEN trên ứng dụng Web...	30
Figure 30: Mật khẩu của sinh viên đã được mã hóa.....	30

# NỘI DUNG BÁO CÁO

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Sơ lược ứng dụng:

Ứng dụng của nhóm là một ứng dụng Quản Lý Điểm Sinh Viên. Mục đích sử dụng hình thức Web quản lý là trong kết cấu dữ liệu của Web bao gồm các trường dữ liệu về danh sách nhân viên, sinh viên, tài khoản, điểm thi... là những dữ liệu quan trọng, và có thể bị đánh cắp. Vì vậy nhóm sẽ tập trung vào phần dữ liệu này để xây dựng các hình thức bảo mật, nâng cao tính an toàn cho ứng dụng.

- Ngôn ngữ - Database:
  - + Ngôn ngữ : Java
  - + Framework: Java Spring MVC
  - + Database : MSSQL
- Các đối tượng dữ liệu trong ứng dụng:
  - + Tài khoản và thông tin người dùng (Accounts)
  - + Thông tin về nhân viên, sinh viên
  - + Điểm sinh viên

### 1.2. Các đối tượng cần bảo mật trong ứng dụng:

#### 1.2.1. Account

Đây là đối tượng dữ liệu tối quan trọng trong các ứng dụng Web, là dữ liệu cần bảo mật tốt nhất trong ứng dụng vì các tài khoản này có thể chỉnh sửa thông tin. Khi tài khoản người dùng bị lấy cắp, sẽ ảnh hưởng đến các thông tin quan trọng của người dùng, và còn có thể gây nguy hại tới hoạt động của ứng dụng.

Để bảo mật trường thông tin này, nhóm sẽ làm tác vụ là hash mật khẩu người dùng thành một chuỗi ký tự phức tạp, để phía tấn công không thể dễ dàng lấy được thông tin tài khoản.

### 1.2.2. Cơ sở dữ liệu (Database):

Cơ sở dữ liệu là nơi ko chỉ lưu trữ các dữ liệu về người dùng mà còn lưu trữ thông tin điểm thi. Các dữ liệu tại đây cũng phải bảo mật một cách cẩn thận vì khi phía tấn công có thể truy cập vào cơ sở dữ liệu, họ sẽ chiếm được quyền điều khiển toàn bộ ứng dụng và hệ thống.

## CHƯƠNG 2: THỰC NGHIỆM

### 2.1. An Toàn Ứng Dụng Web

#### 2.1.1. Chèn mã HTML và Cross-site Scripting (XSS)

##### 2.1.1.1. Vị trí có thể bị chèn mã XSS

- Nguyên nhân chính trang web bị dính lỗ hổng XSS là do ứng dụng web đã không filter kỹ và tin tưởng tuyệt đối vào input do người dùng nhập vào. Malicious user có thể lợi dụng điều này để chèn các mã javascript để thực hiện những hành vi xấu, tiêu biểu là đánh cắp cookie và giả danh người dùng.
- Các dạng XSS: stored XSS, reflected XSS, dom-base XSS.
- Ở trang web, chỉ có chức năng thêm, sửa thông tin sinh viên, nhân viên và điểm là nhận các input từ người dùng và lưu chúng vào database, dễ dàng bị stored XSS.

##### 2.1.1.2. Phòng chống

- Sử dụng hàm `escapeXml` có sẵn trong thư viện jstl để encode các thẻ `<script></script>`

### 2.1.1.3. Thử nghiệm

Tables

DataTables is a third party plugin that is used to generate the demo table below. For more information about DataTables, please visit the [official DataTables documentation](#).

DataTables Add

Show 10 entries Search:

Mã sinh viên	Họ tên	Ngày sinh	Địa chỉ	Mã lớp	Tên đăng nhập	
N19DCAT029	Nguyễn Văn A	2021-11-28	somewhere	D18CQAT01-N	N18DCAT029	Edit
N19DCAT084	asdasdsa<script>alert(document.cookie);</script>	2001-11-11	A	D18CQAT01-N	sv1	Edit
Mã sinh viên	Họ tên	Ngày sinh	Địa chỉ	Mã lớp	Tên đăng nhập	

Showing 1 to 2 of 2 entries

Previous 1 Next

Figure 1: Thử nghiệm XSS

## 2.1.2. SQL injection

### 2.1.2.1. Vị trí có thể chèn mã SQL injection


- SQL injection là hành vi tiêm các câu lệnh sql độc hại để thực hiện hành vi xấu như bypass login, thêm sửa xóa Database, Remote Code Execution...
- Các chức năng có thể bị SQL injection: Login, thêm sửa xóa thông tin sinh viên, nhân viên, điểm.

### 2.1.2.2. Phòng chống

- Sử dụng các PreparedStatement. PreparedStatement là một interface con của Statement. Nó được sử dụng để thực thi các truy vấn được tham số hóa. Khi chúng ta truyền các tham số cho các values, giá trị của nó sẽ được thiết lập bằng việc gọi phương thức setter (setShort, setString...) của PreparedStatement.
- Các phương thức setter của PreparedStatement coi các giá trị truyền vào chỉ có thể là một value, nó sẽ loại bỏ hoàn toàn các ký tự lạ mà người dùng nhập vào trái phép. Do đó loại bỏ mối nguy hại của SQL Injection.
- Mã hoá Password lưu trong cơ sở dữ liệu.

### 2.1.2.3. Thử nghiệm

Welcome Back!

☒ I'm not a robot   
reCAPTCHA  
[Privacy](#) - [Terms](#)

☐ Remember Me

Login

[Forgot Password?](#)

[Create an Account!](#)

Figure 2: Thực hiện SQLi để bypass login

```
ACCOUNT [username=admin' or 1=1 -- , password=a, email=null, sdt=null]
preStmt: SQLServerPreparedStatement:19
result: 0
```

Figure 3: Kết quả trả về của PreparedStatement là 0 (null)

100 % <							
Results		Messages					
MANV	HOTEN	EMAIL	LUONG	TENDN	MATKHAU	PUBKEY	
1	NV1	Nhân viên 1	nhanvien1@mail.com	0x7FA2228811D9EC8127669D4C715447B2CDE4C6F11EEDA0...	NV1	0x7B21848AC9AF35BE0DD82D6B9FC3851934DB8420	RSA/publicKey_NV1
2	nv2	nv2	nv2@gg.com	0x8E18FE6C55F7BD8E95B2C0A8C09F942A8176BA3C573BDB...	nv2	0x981A877ED2BD3412BAAD678ADB8C08107EBF653BD	RSA/publicKey_nv2

Figure 4: Mã hoá mật khẩu lưu trong cơ sở dữ liệu

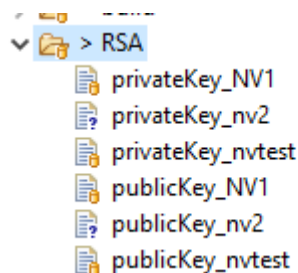


Figure 5: Mã hoá RSA trên Server

### **2.1.3. Authentication Attack**


#### **2.1.3.1. Vị trí có thể bị lỗi Authentication**

- Authentication (xác thực) là một hành động nhằm thiết lập hoặc chứng thực một cái gì đó (hoặc một người nào đó) đáng tin cậy, từ đó được cung cấp các quyền lợi, truy vấn tương ứng với vật/người đã được xác thực. Sau khi bạn được xác thực, hệ thống sẽ biết người đang sử dụng tài khoản/dịch vụ đó chính là bạn.
- Lỗi hỏng xác thực thường xuất hiện khi hệ thống chứa cơ chế xác thực lỏng lẻo, do người lập trình chưa được tiếp cận với các vấn đề an toàn lập trình dẫn đến kẻ tấn công có thể dễ dàng vượt qua (bypass) các lỗi logic hoặc ngoại lệ không mong muốn (unintended problems) trong cơ chế xác thực của hệ thống.
- Một số loại lỗi hỏng xác thực:
  - Lỗi hỏng trong xác thực mật khẩu (password): sử dụng kiểu tấn công brute-force, dựa vào thông tin phản hồi...
  - Lỗi hỏng trong xác thực đa yếu tố (multi-factor authentication): bypass 2FA...
  - Lỗi hỏng qua các cách xác thực khác: cookie dễ đoán và dễ bị bẻ khoá, chúng năng reset password...

#### **2.1.3.2. Phòng chống**

- Sử dụng Captcha để chống brute-force

Welcome Back!

☐ I'm not a robot   
reCAPTCHA  
Privacy - Terms

☐ Remember Me

Login

[Forgot Password?](#)

[Create an Account!](#)

Figure 6: Thử nghiệm Captcha

- Thông tin phản hồi không chứa thông tin dễ đoán, hoặc không phản hồi về trường nào bị sai...
- Password phải được sử dụng có độ dài ít nhất 8 ký tự, chữ, chữ số, kí tự đặc biệt.

#### 2.1.4. Logic Attack

- Ứng dụng Web sử dụng Framework Java Spring MVC, nên với mỗi yêu cầu POST/GET sẽ có các xử lý khác nhau.

```
26 @WebServlet("/sinh-vien")
27 public class SinhVienController extends HttpServlet {
28     private static final long serialVersionUID = 1L;
29     private SinhVienDao sinhVienDao;
30     private LopDao lopDao;
31
32     public SinhVienController() {
33         sinhVienDao = new SinhVienDao();
34         lopDao = new LopDao();
35     }
36
37     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
38
39     }
40
41     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
42
43     }
44 }
```

Figure 7: Xử lý request GET/POST với endpoint sinh-vien

- Các Endpoint yêu cầu trước đó phải xác thực mới truy cập được.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    RequestDispatcher dispatcher = this.getServletContext().getRequestDispatcher("/sinhvien.jsp");
    HttpSession session = request.getSession();
    ACCOUNT ac = session != null ? (ACCOUNT) session.getAttribute("userlogin") : new ACCOUNT();
    if (ac == null) {
        response.sendRedirect("./login");
    }
    else {
        List<SinhVien> svcs = sinhVienDao.getAllSinhVien();
        List<Lop> lops = lopDao.getAllLop();
        request.setAttribute("svcs", svcs);
        request.setAttribute("lops", lops);
        dispatcher.forward(request, response);
    }
}
```

Figure 8: Người dùng phải đăng nhập trước đó để có session truy cập vào các endpoint

- Không tin tưởng tuyệt đối vào Input của Users.

```
<c:forEach var="sv" items="${svcs }">
    <tr>
        <td>${fn:escapeXml(sv.maSV) }</td>
        <td>${fn:escapeXml(sv.hoTen) }</td>
        <td>${fn:escapeXml(sv.ngaySinh) }</td>
        <td>${fn:escapeXml(sv.diaChi) }</td>
        <td>${fn:escapeXml(sv.maLop) }</td>
        <td>${fn:escapeXml(sv.tenDN) }</td>
        <td>
            <a href="edit-sinh-vien?msv=${fn:escapeXml(sv.maSV) }">Edit</a>
        </td>
    </tr>
</c:forEach>
```

Figure 9: HTML entity các giá trị

- Sử dụng mã hoá RSA256

### 2.1.5. Tấn công vào trình duyệt web và sự riêng tư của người dùng

- Các giải pháp: Khi đưa ứng dụng lên môi trường product, nên sử dụng thêm WAF như cloudflare, chứng chỉ SSL...


### 2.1.6. Xác thực người dùng và trao quyền truy cập (Access control)

- Yêu cầu người dùng đăng nhập bằng username/password.



## Welcome Back!

☐ I'm not a robot
 


  
reCAPTCHA  
Privacy - Terms

☐ Remember Me

Figure 10: Authentication

- Cấu hình đầy đủ Authentication, các endpoint trong ứng dụng Web đều yêu cầu Authentication để có thể truy cập và thay đổi.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    RequestDispatcher dispatcher = this.getServletContext().getRequestDispatcher("/success.jsp");

    HttpSession session = request.getSession();
    ACCOUNT ac = session != null ? (ACCOUNT) session.getAttribute("userlogin") : new ACCOUNT();
    if (ac == null) {
        response.sendRedirect("./login");
    }
    else {
        dispatcher.forward(request, response);
    }
    dispatcher.forward(request, response);
}
}
```

Figure 11: Người dùng phải login thì mới có thể có cooke để truy cập vào các endpoint

DataTables
Add

Chỉ có admin mới được thêm mới nhân viên

Figure 12: Nhân viên thì sẽ không thể thêm nhân viên mới

**Tables**

DataTables is a third party plugin that is used to generate the demo table below. For more information about DataTables, please visit the [official DataTables documentation](#).

DataTables [Add](#)

Mã nhân viên

Họ tên nhân viên

Email

Lương

Tên đăng nhập

Mật khẩu

Nhập lại mật khẩu

[Thêm](#)

Figure 13: Admin có thêm nhân viên mới

### 2.1.7. Bảo mật phiên làm việc

- Các token được sinh ra ngẫu nhiên trên Framework Java Spring MVC, có độ dài lớn và khó đoán, khó thực hiện brute-force, không phụ thuộc vào các yếu tố khác.
- Không để token của phiên vào URL.
- Với Java Spring MVC, sau một thời gian người dùng không hoạt động sẽ tự động huỷ token.

Application						
Filter						
Application	Name	Value	Domain	Path	Expires ...	Size
	uuid	f469d2edc09c64ab73a86db666293a965570294	.unspla...	/	2023-1...	43
	JSESSIONID	6A5715B5E773E5F8304EABFEE013FB81	localhost	/atweb	Session	42

Figure 14: Bảo mật phiên làm việc

- Ứng dụng Web có tính năng Log-Out, và session-cookie phiên đó sẽ bị huỷ và không thể sử dụng lại.

```

@WebServlet("/logout")
public class logout extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public logout() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession();
        session.removeAttribute("userlogin");
        session.invalidate();
        response.sendRedirect("./login");
    }
}

```

Figure 15: Tính năng Log-Out

### 2.1.8. Bảo vệ máy chủ web

- Ứng dụng Web được xây dựng trên Framework Java Spring MVC nên máy chủ chỉ xử lý các Request đến các Endpoint được chỉ định, và các Endpoint không được chỉ định trước đó sẽ không thể truy cập. Do đó Attacker sẽ không thể duyệt các đường dẫn.



Figure 16: Bảo mật máy chủ Web

- Cũng như không có các tài khoản/nội dung ngầm định, không có chế độ debug.
- Chỉ định những phương thức nào sẽ được đưa vào những hàm nào xử lý.

### 2.1.9. Bảo mật hệ thống file

- Các chức năng trong ứng dụng Web Quản Lý Điểm Sinh Viên đều là chức năng nội bộ, yêu cầu có Username & Password với được truy cập.
- Các quyền của mỗi User được giới hạn nhất định.

- Trang Web không có bất cứ file backup nào, không public source.
- Với Java Spring MVC thì không thể liệt kê và duyệt các thư mục.

## **2.2. An Toàn Cơ Sở Dữ Liệu**

### **2.2.1. Default and Weak Passwords**

#### **2.2.1.1. Mô tả**

- Trong quá trình cài đặt các hệ Quản trị Cơ sở dữ liệu thì sẽ có nhiều tài khoản được tạo mặc định ở trong các DBMS. Những tài khoản đó được tạo những mật khẩu mặc định hoặc không có mật khẩu. Điều này dẫn đến khi các kẻ tấn công kết nối vào được các cổng của CSDL dùng các mật khẩu mặc định hoặc là có thể dò tìm ra tài khoản đăng nhập vào được CSDL thực hiện hành vi phá hoại
- VD:
  - Oracle: User: system / Password: manager
  - MySQL: User: root /Password: null
  - SQL Server: User: SA /Password: null

#### **2.2.1.2. Phòng chống**

- Đặt hoặc thay đổi các mật khẩu mặc định cho các tài khoản tạo mặc định trong các CSDL. Mật khẩu phải đảm bảo độ dài, độ phức tạp để đảm bảo các kẻ tấn công không thể dễ dàng dò tìm ra được.

#### **2.2.1.3. Thử nghiệm**

- Sử dụng SQL Server thay đổi mật khẩu mạnh cho tài khoản SA.

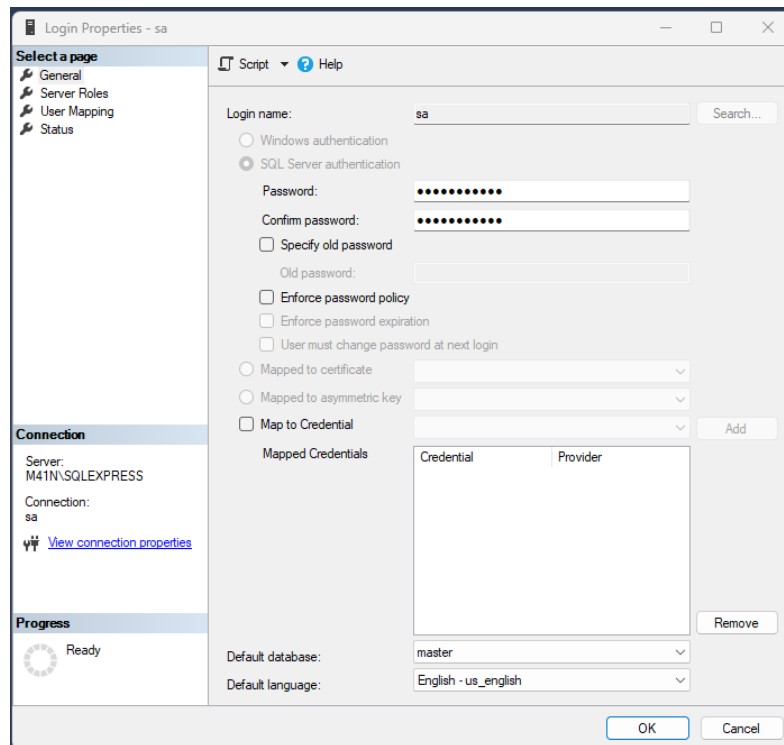


Figure 17: Thay đổi mật khẩu mạnh cho tài khoản mặc định SA

```

public class ConnectionToDB {
    public ConnectionToDB() {
    }

    public static Connection getConnect() {
        Connection connection = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            String URL = "jdbc:sqlserver://localhost:1433;databaseName=FINALLTM; user=sa; password=ptithcm@123";
            Connection con = DriverManager.getConnection(URL);
            return con;
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return connection;
    }
}

```

Figure 18: Kết nối trong ứng dụng web

## 2.2.2. SQL Injection in the DBMS ( Lỗi chèn mã SQL)

### 2.2.2.1. Mô tả

- Là một lỗ hổng cực kì nghiêm trọng khi Input từ ứng dụng Web truyền đến Database không được kiểm tra chặt chẽ cho kẻ tấn công thực thi câu truy vấn ngay trên ứng dụng Web lên Database thực hiện các hành vi phá hoại như: chèn, sửa, xóa dữ liệu, đánh cắp thông tin trong CSDL hoặc chiếm quyền điều khiển hệ thống máy chủ CSDL.

### 2.2.2.2. Phòng chống

- Phòng chống trên ứng dụng Web. Không được truyền các biến trực tiếp vào các câu truy vấn. Nên sử dụng các Framework cho chức năng chống SQL injection.

### 2.2.2.3. Thử nghiệm

- Ứng dụng Web phòng chống SQL Injection bằng cách khi sử dụng câu truy vấn thông qua thư viện PreparedStatement. Kẻ tấn công không thể thực hiện câu truy vấn độc hại được.

```
public int check(ACCOUNT account) {  
    String CHECK_USER_SQL = "select username from ACCOUNT where username = ? and password = ?";  
    int result = 0;  
  
    // kết nối db  
    Connection connection = new ConnectionToDB().getConnection();  
    try {  
        PreparedStatement preStmt = connection.prepareStatement(CHECK_USER_SQL);  
        preStmt.setString(1, account.getUsername());  
        preStmt.setString(2, account.getPassword());  
  
        System.out.println("preStmt: " + preStmt.toString());  
        // thực hiện thêm account vào  
        ResultSet rs = preStmt.executeQuery();  
  
        if (rs.next()) {  
            result = 1;  
            System.out.println("numrow = " + rs.getRow());  
        }  
        else  
            result = 0;  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        return 0;  
    }  
}
```

Figure 19: Sử dụng thư viện để thực hiện câu truy vấn chống SQLi

## 2.2.3. Excessive User & Group Privileges

### 2.2.3.1. Mô tả

- Lỗi hỏng này xuất hiện khi có nhiều người dùng hoặc nhóm người dùng được cấp quyền nhập cao quá mức cần thiết so với các công việc được giao dẫn tới lạm dụng quyền và khi kẻ tấn công có được tài khoản có quyền hạn cao có thể thực hiện những hành vi nguy hiểm.
- Trên thực tế nhiều người dùng CSDL chỉ được tạo để truy cập và nhập dữ liệu thôi mà được cấp quyền quản trị hoặc là chủ sở hữu CSDL. Người dùng đó có thể thực hiện các thao tác cao khác như : xóa, sửa, thêm table, xóa table,...

### 2.2.3.2. Phòng chống

- Xem xét cấp quyền cho người dùng đúng với quyền hạn của người dùng đó. Thường xuyên kiểm soát, rà soát các người dùng có thực hiện đúng với quyền hạn trên CSDL.

### 2.2.3.3. Thử nghiệm

- Cấp quyền cho user DBRead chỉ có quyền truy xuất dữ liệu trên Database QLDSV.

The screenshot shows the 'Login - New' dialog box in SQL Server Enterprise Manager. The 'General' tab is active. The 'Login name' field contains 'DBRead'. The 'Authentication' section has 'SQL Server authentication' selected. The 'Password' and 'Confirm password' fields are filled with masked characters. The 'Enforce password policy' section has three checkboxes checked: 'Enforce password policy', 'Enforce password expiration', and 'User must change password at next login'. The 'Mapped Credentials' section is empty. The 'Default database' dropdown is set to 'QLDSV'. The 'Default language' dropdown is set to 'Default'. The 'OK' and 'Cancel' buttons are at the bottom right.

Figure 20: Tạo User DBRead chỉ có quyền trên database QLDSV

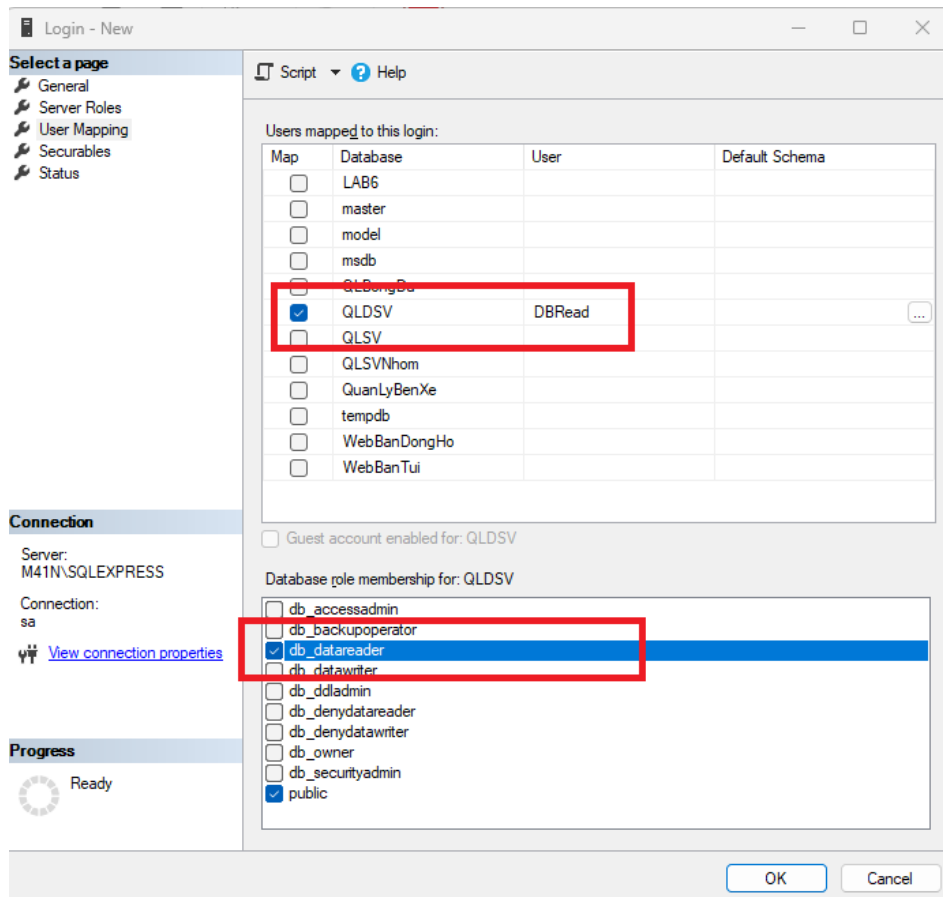


Figure 21: User chỉ có quyền đọc trong Database QLDSV



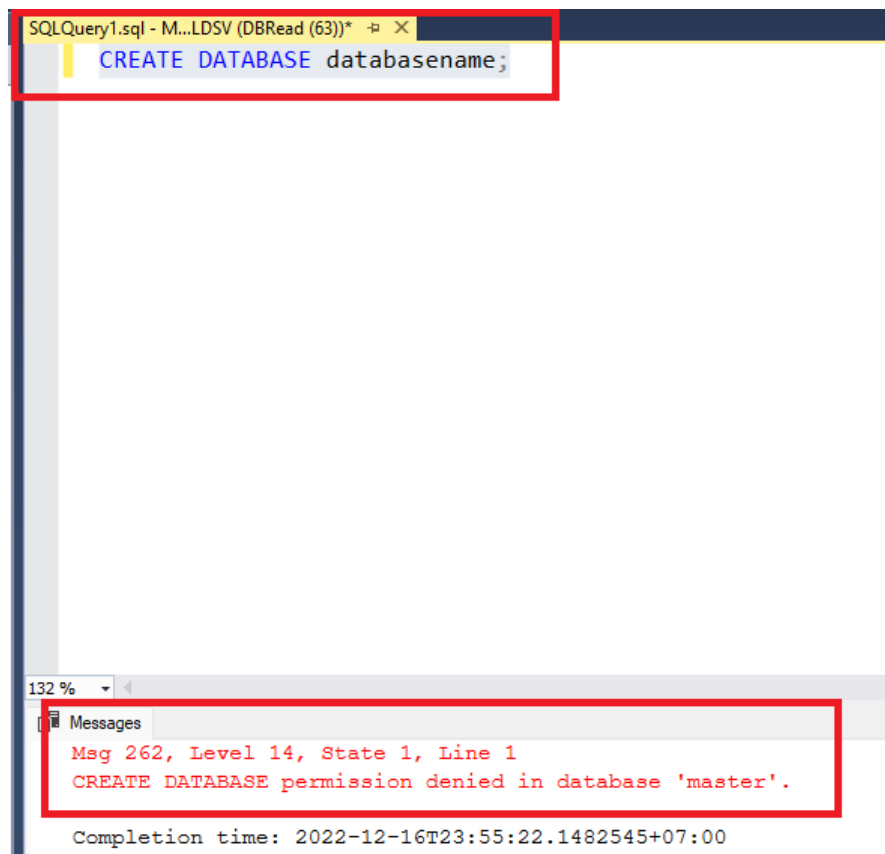


Figure 22: Thử trường hợp User DBRead muốn quyền Tạo Database

## 2.2.4. Unnecessary Enabled DBMS Features

### 2.2.4.1. Mô tả

- Trong các DBMS có những tính năng nâng cao có thể tương tác trực tiếp với hệ điều hành mà người dùng quản trị không cần thiết. Khi người quản trị không tắt các tính năng này. Điều này dẫn đến kẻ tấn công khi có được tài khoản DBMS có thể sử dụng các tính năng để làm tổn hại đến cả hệ thống hoặc chiếm cả quyền kiểm soát hệ điều hành.
- Ví dụ các tính năng nâng cao trong các DBMS:
  - Oracle: UTL\_FILE cho phép người dùng có đặc quyền hệ thống tạo hoặc thay đổi đối tượng DIRECTORY.
  - MySQL: cho phép quyền trên User Table (mysql.user)
  - SQL Server: OLEDB Ad Hoc Query – OPENROWSET, xp\_cmdshell : cho phép thực hiện command lên hệ điều hành

### 2.2.4.2. Phòng chống

- Tắt các tính năng mặc định không cần thiết của DBMS. Phân quyền cho các user hợp lý đúng với chức năng.

### 2.2.4.3. Thử nghiệm

- Khi chưa tắt tính năng xp\_cmdshell thì người dùng có đặc quyền có thể thực thi các OS command ngay trên SQL Server. Đây là một tính năng không cần thiết cho một người quản trị.

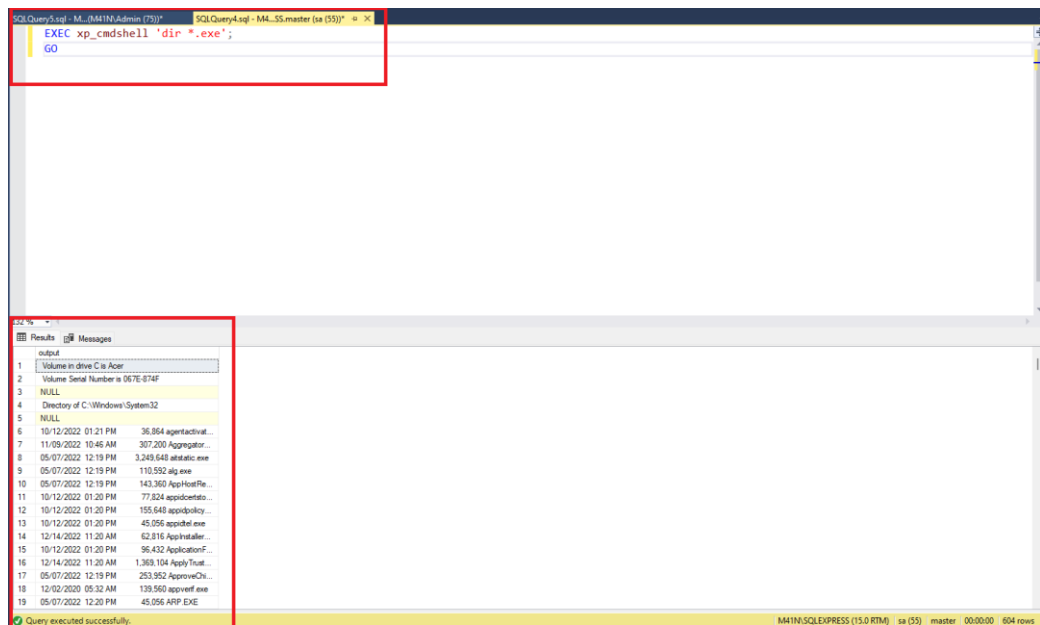


Figure 23: Thực hiện câu lệnh "dir \*.exe" xuất danh sách file trên SQL Server

- Tắt tính năng không cần thiết xp\_cmdshell để không kiểm soát được hệ thống OS.

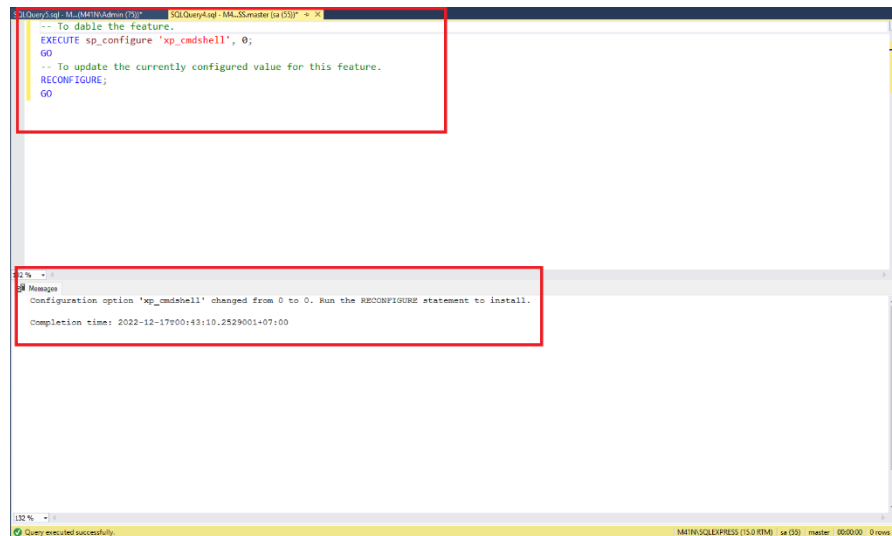


Figure 24: Tắt cấu hình cho phép "Show Advanced Option" để tắt xp\_cmdshell

- Sau khi tắt xp\_cmdshell không thể thực hiện câu lệnh trên hệ điều hành.

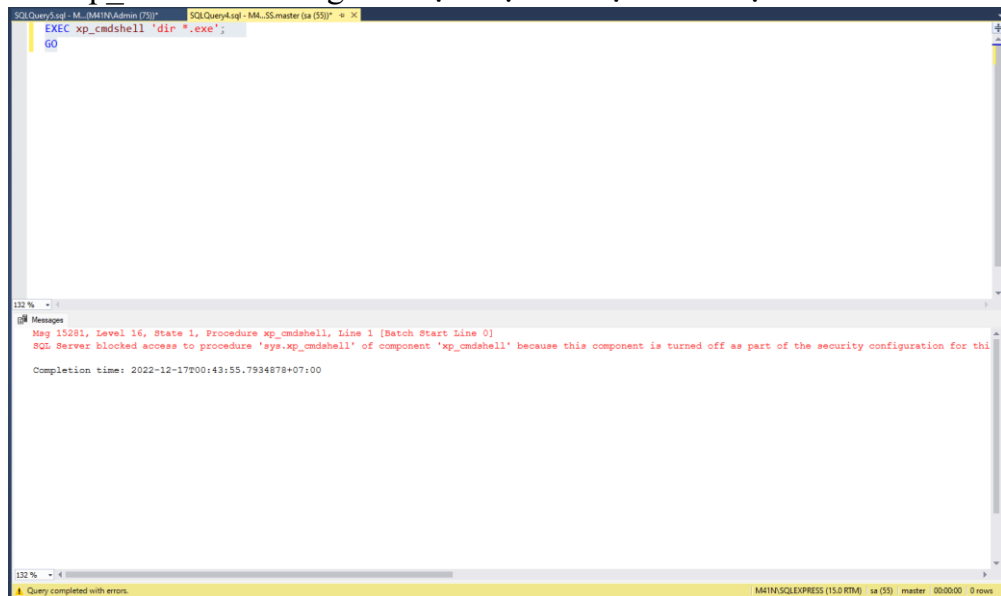


Figure 25: Không thực hiện OS Command sau khi tắt xp\_cmdshell

## 2.2.5. Unpathed Database

### 2.2.5.1. Mô tả

- Mỗi ngày đều có những chuyên gia bảo mật cũng như nghiên cứu các lỗ hổng của các ứng dụng lớn trong đó có các CSDL. Và các nhà cung cấp cũng cho phép điều đó để khám phá ra lỗ hổng của ứng dụng của họ. Sau khi họ vá thì các lỗ hổng sẽ được công bố trên Internet như là CVE, 0-day, 1-day,... Điều này dẫn

đến khi các CSDL bị dính lỗi đã công bố thì kẻ tấn công có thể dựa theo hướng dẫn để khai thác lỗ hổng đó.

### 2.2.5.2. Phòng chống

- Các CSDL hoặc các module kèm theo phải được vá lỗi hoặc cập nhật phù hợp lên các phiên bản mới để những kẻ tấn công không thể khai thác các lỗ hổng đã biết.

### 2.2.5.3. Thử nghiệm

- Các lỗ hổng của SQL Server 2000 có rất nhiều CVE đã được công bố trong số đó có lỗi cực kì nguy hiểm có thể khai thác trực tiếp vào hệ thống người dùng.

Microsoft » <a href="#">Sql Server</a> » 2000 SP2 : Security Vulnerabilities														
Cpe Name:cpe:2.3:a:microsoft:sql_server:2000:sp2:*:*:*:*:*:*														
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9														
Sort Results By : <a href="#">CVE Number Descending</a> <a href="#">CVE Number Ascending</a> <a href="#">CVSS Score Descending</a> <a href="#">Number Of Exploits Descending</a>														
<a href="#">Covv Results</a> <a href="#">Download Results</a>														
#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2003-0232</a>			Exec Code Overflow	2003-06-27	2018-10-12	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Microsoft SQL Server 7, 2000, and MSDE allows local users to execute arbitrary code via a certain request to the Local Procedure Calls (LPC) port that leads to a buffer overflow.														
2	<a href="#">CVE-2003-0231</a>			DoS	2003-08-27	2018-10-12	5.0	None	Remote	Low	Not required	None	None	Partial
Microsoft SQL Server 7, 2000, and MSDE allows local or remote authenticated users to cause a denial of service (crash or hang) via a long request to a named pipe.														
3	<a href="#">CVE-2003-0230</a>	264		+Priv	2003-08-27	2018-10-12	7.2	None	Local	Low	Not required	Complete	Complete	Complete
Microsoft SQL Server 7, 2000, and MSDE allows local users to gain privileges by hijacking a named pipe during the authentication of another user, aka the "Named Pipe Hijacking" vulnerability.														
4	<a href="#">CVE-2002-1981</a>				2002-12-31	2008-09-05	5.0	None	Remote	Low	Not required	None	Partial	None
Microsoft SQL Server 2000 through SQL Server 2000 SP2 allows the "public" role to execute the (1) sp_MSsetServerProperties or (2) sp_MSsetalertinfo stored procedures, which allows attackers to modify configuration including SQL server startup and alert settings.														
5	<a href="#">CVE-2002-1872</a>				2002-12-31	2008-09-05	5.0	None	Remote	Low	Not required	Partial	None	None
Microsoft SQL Server 6.0 through 2000, with SQL Authentication enabled, uses weak password encryption (XOR), which allows remote attackers to sniff and decrypt the password.														
6	<a href="#">CVE-2002-1145</a>			+Priv	2002-10-28	2018-10-12	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
The xp_runwebtask stored procedure in the Web Tasks component of Microsoft SQL Server 7.0 and 2000, Microsoft Data Engine (MSDE) 1.0, and Microsoft Desktop Engine (MSDE) 2000 can be executed by PUBLIC, which allows an attacker to gain privileges by updating a webtask that is owned by the database owner through the msdb.dbo.mswebtasks table, which does not have strong permissions.														
7	<a href="#">CVE-2002-1138</a>				2002-10-11	2018-10-12	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Microsoft SQL Server 7.0 and 2000, including Microsoft Data Engine (MSDE) 1.0 and Microsoft Desktop Engine (MSDE) 2000, writes output files for scheduled jobs under its own privileges instead of the entity that launched it, which allows attackers to overwrite system files, aka "Flaw in Output File Handling for Scheduled Jobs."														
8	<a href="#">CVE-2002-1137</a>			Exec Code Overflow	2002-10-11	2018-10-12	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Buffer overflow in the Database Console Command (DBCC) that handles user inputs in Microsoft SQL Server 7.0 and 2000, including Microsoft Data Engine (MSDE) 1.0 and Microsoft Desktop Engine (MSDE) 2000, allows attackers to execute arbitrary code via a long SourceDB argument in a "non-SQL OLEDB data source" such as FoxPro, a variant of CAN-2002-0644.														
9	<a href="#">CVE-2002-1123</a>			Exec Code Overflow	2002-09-24	2018-10-12	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Buffer overflow in the authentication function for Microsoft SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000 allows remote attackers to execute arbitrary code via a long request to TCP port 1433, aka the "Hello" overflow.														
10	<a href="#">CVE-2002-0982</a>			Exec Code	2002-09-24	2016-10-18	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Microsoft SQL Server 2000 SP2, when configured as a distributor, allows attackers to execute arbitrary code via the @scriptfile parameter to the sp_MScopyscript stored procedure.														
11	<a href="#">CVE-2002-0859</a>			Exec Code Overflow	2002-09-05	2018-08-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Buffer overflow in the OpenDataSource function of the Jet engine on Microsoft SQL Server 2000 allows remote attackers to execute arbitrary code.														
12	<a href="#">CVE-2002-0721</a>				2002-09-05	2018-10-12	10.0	None	Remote	Low	Not required	Complete	Complete	Complete
Microsoft SQL Server 7.0 and 2000 installs with weak permissions for extended stored procedures that are associated with helper functions, which could allow unprivileged users, and possibly remote attackers, to run stored procedures with administrator privileges via (1) xp_execresultset, (2) xp_printstatements, or (3) xp_displayparamstmt.														
13	<a href="#">CVE-2002-0650</a>			DoS	2002-08-12	2018-10-12	5.0	None	Remote	Low	Not required	None	None	Partial
The keep-alive mechanism for Microsoft SQL Server 2000 allows remote attackers to cause a denial of service (bandwidth consumption) via a "ping" style packet to the Resolution Service (UDP port 1434) with a spoofed IP address of another SQL Server system, which causes the two servers to exchange packets in an infinite loop.														
14	<a href="#">CVE-2002-0649</a>	119		DoS Exec Code	2002-08-12	2018-10-19	7.5	None	Remote	Low	Not required	Partial	Partial	Partial

Figure 26: Danh sách các CVE của SQL Server 2000

- ➔ Khi hệ thống còn sử dụng phiên bản này thì sẽ có nhiều rủi ro bị hacker tấn công.
- Phải cập nhật lên các phiên bản mới của SQL Server để khắc phục được các lỗ hổng đó. Ở đây chúng em sử dụng SQL Server Studio 18 có ít CVE đã được công bố và tác động là nhỏ không gây ảnh hưởng tới hệ thống.

Microsoft » [Sql Server Management Studio](#) » [18.3.1](#) : Security Vulnerabilities

Cpe Name: cpe:2.3:a:microsoft:sql\_server\_management\_studio:18.3.1:\*:\*:\*:\*:\*:\*

CVSS Scores Greater Than: [0](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#)

Sort Results By: [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Covv Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2019-1376</a>	<a href="#">755</a>			2019-10-10	2020-08-24	4.0	None	Remote	Low	???	Partial	None	None
An information disclosure vulnerability exists in Microsoft SQL Server Management Studio (SSMS) when it improperly enforces permissions, aka 'SQL Server Management Studio Information Disclosure Vulnerability'. This CVE ID is unique from CVE-2019-1313.														
2	<a href="#">CVE-2019-1313</a>	<a href="#">755</a>			2019-10-10	2020-08-24	4.0	None	Remote	Low	???	Partial	None	None
An information disclosure vulnerability exists in Microsoft SQL Server Management Studio (SSMS) when it improperly enforces permissions, aka 'SQL Server Management Studio Information Disclosure Vulnerability'. This CVE ID is unique from CVE-2019-1376.														

Total number of vulnerabilities: 2 Page: [1](#) (This Page)

Figure 27: Danh sách các CVE của SQL Server Studio 18

## 2.2.6. Unencrypted Data

### 2.2.6.1. Mô tả

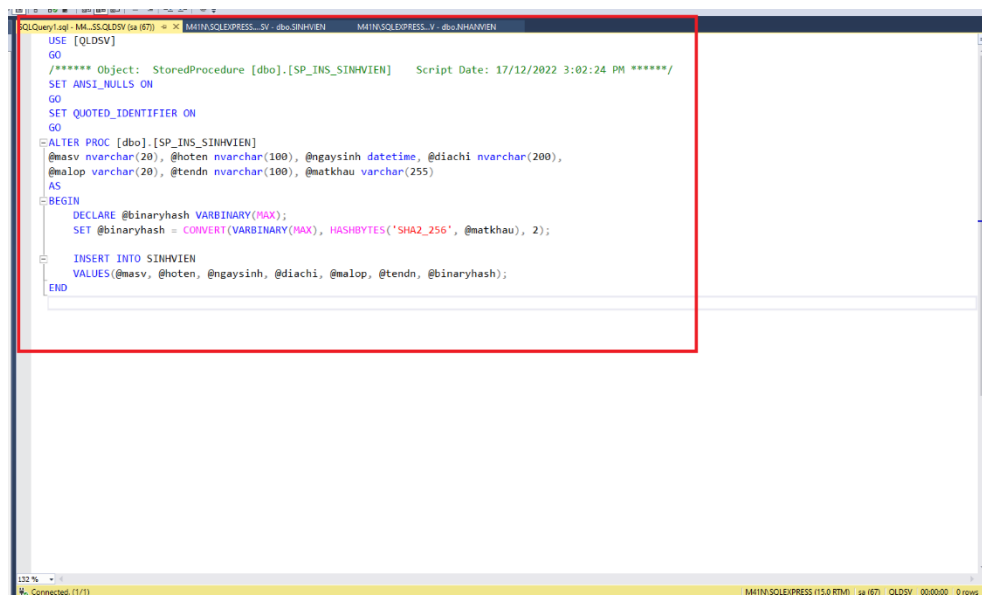
- Các dữ liệu nhạy cảm như tài khoản, thông tin người dùng,... trong quá trình truyền dữ liệu hoặc là nhập vào thì không được mã hóa. Điều này dẫn đến các kẻ tấn công có thể nghe trộm thông tin bằng cách bắt đường truyền, hoặc là đánh cắp dữ liệu trên database gây thiệt hại cho ứng dụng.

### 2.2.6.2. Phòng chống

- Đối với dữ liệu nhập trên database thì có thể lưu dữ liệu ở dạng mã hóa để kẻ tấn công lấy được nhưng không thể đọc nội dung.
- Dữ liệu trên đường truyền cần có sử dụng các giao thức bảo mật để ngăn chặn kẻ tấn công có thể nghe trộm được dữ liệu như là: sử dụng SSL/TLS, Kerberos, Oracle ASO...

### 2.2.6.3. Thử nghiệm

- Trong quá trình lưu dữ liệu thì những dữ liệu nhạy cảm như mật khẩu của người dùng... Những dữ liệu nhạy cảm này nên được lưu dưới dạng mã hóa để không bị đánh cắp.
- Sử dụng SP\_INS\_SINH\_VIEN để mã hóa dữ liệu SHA256 dữ liệu trước khi lưu vào CSDL.



```
USE [QLDSV]
GO
/***** Object: StoredProcedure [dbo].[SP_INS_SINHVIEN]    Script Date: 17/12/2022 3:02:24 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROC [dbo].[SP_INS_SINHVIEN]
    @masv nvarchar(20), @hoten nvarchar(100), @ngaysinh datetime, @diachi nvarchar(200),
    @malop varchar(20), @tendn nvarchar(100), @matkhou varchar(255)
AS
BEGIN
    DECLARE @binaryhash VARBINARY(MAX);
    SET @binaryhash = CONVERT(VARBINARY(MAX), HASHBYTES('SHA2_256', @matkhou), 2);

    INSERT INTO SINHVIEN
    VALUES(@masv, @hoten, @ngaysinh, @diachi, @malop, @tendn, @binaryhash);
END
```

Figure 28: SP để thêm sinh viên có mật khẩu đã được mã hóa

```

1 package atweb.nhom6.dao;
2
3 import java.sql.CallableStatement;
4
5 public class SinhVienDao {
6
7     // Mã hóa phía database
8     public void insertSinhVien(SinhVien sv) {
9         Connection conn = ConnectionToDB.getConnection();
10        // MASV, HOTEN, NGAYSINH, DIACHI, MALOP, TENDN, MATKHAU
11        String sql = "{call SP_INS_SINHVIEN(?, ?, ?, ?, ?, ?, ?)}";
12        try {
13            CallableStatement callableStatement = conn.prepareCall(sql);
14            callableStatement.setString(1, sv.getMaSV());
15            callableStatement.setString(2, sv.getHoTen());
16            java.sql.Date sqlNgaySinh = new java.sql.Date(sv.getNgaySinh().getTime());
17            callableStatement.setDate(3, sqlNgaySinh);
18            callableStatement.setString(4, sv.getDiaChi());
19            callableStatement.setString(5, sv.getMaLop());
20            callableStatement.setString(6, sv.getTenDN());
21            callableStatement.setString(7, sv.getMatKhai());
22            callableStatement.execute();
23        } catch (SQLException e) {
24            e.printStackTrace();
25        }
26    }
27
28    public List<SinhVien> getAllSinhVien() {
29        List<SinhVien> sinhViens = new ArrayList<>();
30    }
31
32 }

```

Figure 29: Thực hiện gọi đến SP\_INS\_SINHVIEN trên ứng dụng Web

select \* from SINHVIEN

id	id	id	id	id	id	id	id
N190CAT010	Nguyễn Quốc Huy	2021-11-20 00:00:00.000	nam	nam	nam	nam	nam
N190CAT011	Nguyễn Quang Chính	2001-09-17 00:00:00.000	nam	nam	nam	nam	nam

Figure 30: Mật khẩu của sinh viên đã được mã hóa

## TÀI LIỆU THAM KHẢO

- [1] *Bài Giảng An Toàn Ứng Dụng Web và Cơ Sở Dữ Liệu*, Học Viện Công Nghệ Bưu Chính Viễn Thông, 2017.
- [2] *Bài Giảng An Toàn Ứng Dụng Web và Cơ Sở Dữ Liệu*, Th.S Phan Nghĩa Hiệp, Học Viện Công Nghệ Bưu Chính Viễn Thông, 2021.

## BẢNG PHÂN CÔNG CÔNG VIỆC

	Trần Anh Dũng	Nguyễn Quang Chính	Mai Thế Chuyên Em	Nguyễn Hoàng Đại Nghĩa	Trần Thanh Tra
Code front-end	✓	✓			
Code backend			✓	✓	✓
Chèn HTML, XSS	✓		✓		
SQL injection		✓		✓	
Authen Attack		✓	✓		
Logic Attack			✓		✓
Tấn công trình duyet Web		✓			✓
Access Control	✓	✓			
Bảo mật phiên làm việc				✓	✓
Bảo mật máy chủ web		✓	✓		
Bảo mật hệ thống file	✓			✓	
Default an Weak Passwords			✓	✓	
SQLi DBMS	✓	✓		✓	
Excessive User & Group Privileges	✓		✓		
Unnecessary Enabled DBMS Features				✓	✓
Unpathed Database	✓				✓
Unencryted Data	✓				✓
Tổng hợp	✓	✓			



