1. Các biện pháp phòng chống SQL Injection

1.1. Sử dụng Prepared Statement (Parameterized Query) và Stored procedure

Cách này cho phép định nghĩa cấu trúc câu lệnh SQL trước, trong khi đó dữ liệu người dùng nhập vào được truyền vào dưới dạng tham số riêng biệt và và không bao giờ được coi là một phần của câu lệnh SQL nên kẻ tấn công không thể chèn thêm mã SQL mới vào lệnh, ngay cả khi input có chứa các ký tự hoặc chuỗi độc hại. Cơ chế này đảm bảo rằng cấu trúc truy vấn không thể bị thay đổi bởi dữ liệu đầu vào. Ngoài ra, cơ sở dữ liệu còn tự động kiểm tra và "bind" tham số theo đúng kiểu dữ liệu, giúp đảm bảo an toàn và tăng tính ổn định cho ứng dụng.

Đối với Stored Procedure, sự khác biệt chính giữa prepared statement và stored procedure là stored procedure được lưu trực tiếp trong database và được ứng dụng gọi khi cần.

Ví dụ:

SQL Injection

```
string username = txtUser.Text;

string sql = "SELECT * FROM Users WHERE Username = '" + username + "'";

SqlCommand cmd = new SqlCommand(sql, connection);
SqlDataReader reader = cmd.ExecuteReader();
```

Dùng Prepared Stament:

```
string username = txtUser.Text;

string sql = "SELECT * FROM Users WHERE Username = @username";

SqlCommand cmd = new SqlCommand(sql, connection);
cmd.Parameters.AddWithValue("@username", username);

SqlDataReader reader = cmd.ExecuteReader();
```

1.2. Kiểm Tra Dữ Liệu Đầu Vào (Validation Input)

Biện pháp này đảm bảo rằng dữ liệu người dùng nhập vào tuân theo định dạng, kiểu dữ liệu, và giới hạn kích thước đã được xác định trước. Đặt ra một "whitelisting" về những gì được phép nhập. Hơn nữa nó còn giúp ngăn ngừa SQL Injection bằng cách loại bỏ các ký tự đặc biệt, mã độc hại, hoặc cấu trúc dữ liệu không hợp lệ ngay từ đầu, trước khi chúng được xử lý bởi cơ sở dữ liệu.

Ví dụ: chỉ cho phép số trong ô nhập giá tiền, chỉ cho phép chữ cái trong ô nhập tên, hay giới hạn độ dài của chuỗi.

1.3. Nguyên tắc quyền tối thiểu (Least Privilege)

Nguyên tắc cấp quyền tối thiểu hạn chế quyền truy cập của người dùng xuống mức cần thiết. Nhờ vậy, khả năng thành công của tấn công SQL Injection được giảm đáng kể.

Ví dụ: một người dùng chỉ cần quyền đọc dữ liệu thì không nên được cấp quyền xóa hay chỉnh sửa bản ghi trong cơ sở dữ liệu.

1.4. Web Application Firewall (WAF)

WAF hoạt động như một lớp chắn giữa ứng dụng và Internet, giúp lọc bỏ các yếu tố độc hại. Với công nghệ machine learning, WAF hiện đại có thể phát hiện các mẫu bất thường như hành vi cố gắng thực hiện SQL Injection và chặn chúng trước khi đến được ứng dụng.

2. Các công cụ phát hiện SQL Injectio

Công cụ tự động

Công cụ	**Chức năng**
OWASP ZAP	Quét lỗ hổng web, phát hiện SQLI, XSS…
Burp Suite Scanner	Tự động dò tìm injection trong tham số HTTP.
Acunetix	Công cụ thương mại mạnh về phát hiện SQLI.
Wapiti	Scanner mã nguồn mở, tìm SQL Injection và nhiều lỗ hổng khác.
Nessus / OpenVAS	Công cụ quét bảo mật tổng hợp, có module SQLI.

Kiểm tra thủ công (Manual Code Reviews): có thể phát hiện các lỗ hổng bảo mật phức tạp như lỗi logic hoặc điểm yếu phụ thuộc ngữ cảnh, những thứ thường bị bỏ sót bởi các công cụ tự động. Mặc dù việc này tốn khá nhiều thời gian nhưng nó có thể làm lộ ra các vectơ tấn công tiềm ẩn rất khó phát hiện qua tự động hóa (như việc sử dụng dữ liệu đầu vào không đúng cách). Do đó, biện pháp này không nên thay thế mà cần được tích hợp với các công cụ quét tự động để đạt hiệu quả bảo mật toàn diện nhất.

Phân tích Mối đe dọa (Threat Analysis): là quá trình xác định cách thức kẻ tấn công có thể khai thác lỗ hổng SQL Injection. Bằng cách lập mô hình mối đe dọa (threat modeling) để xem xét kiến trúc ứng dụng, vị trí dữ liệu nhạy cảm. Phân tích này giúp ưu tiên các lỗ hổng cần khắc phục dựa trên tác động tiềm tàng và khả năng bị khai thác. Ngoài ra, nó còn giúp xác định các điểm xâm nhập (entry points) của SQLi, qua đó giúp hệ thống cần tập trung khắc phục phần nào.

3. Bkav – Sự cố tấn công bởi SQL Injection

3.1. Bối cảnh & sự việc

- Vào khoảng tháng 8 / 2021, một cá nhân có biệt danh chunxong đã đăng video mô tả việc tấn công vào hệ thống máy chủ VPN của Bkav, với cáo buộc khai thác lỗ hổng SQL Injection.
- Chunxong tuyên bố rằng quá trình truy cập chỉ mất khoảng 5 phút nhờ lỗ hổng "SQL Injection truyền thống".
- Hình ảnh kèm theo cho thấy dữ liệu và mã nguồn một số sản phẩm Bkav bị rao bán với giá hàng chục nghìn USD.
- Bkav xác nhận rằng có "dữ liệu cũ" bị rò rỉ từ một số module phần mềm, và cho rằng sự việc không ảnh hưởng đến dịch vụ chính thức của công ty.

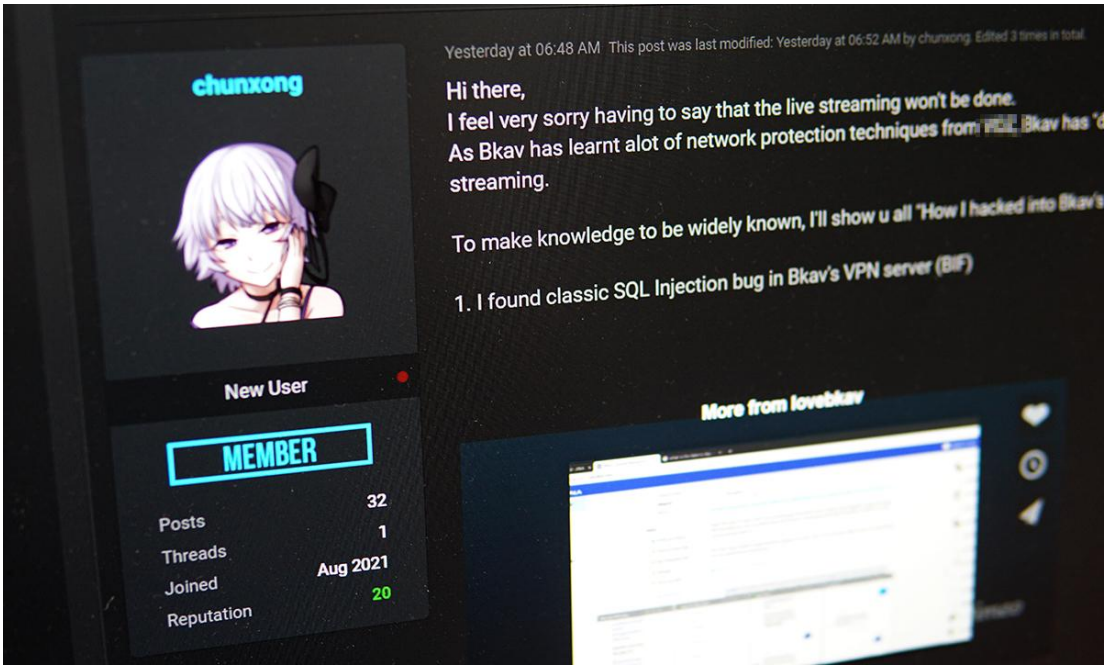

Figure 1. Post of Chunxung on forum.(cre: https://vnexpress.net/hacker-da-tan-cong-bkav-tu-mot-loi-co-ban-4341131.html)

3.2. Nguyên nhân – điểm yếu kỹ thuật

- Lỗ hổng SQL Injection xuất hiện trên máy chủ dịch vụ VPN của Bkav theo lời hacker.
- Theo chuyên gia, điều này cho thấy một hệ thống "đã bị bỏ quên" hoặc chưa được kiểm tra kỹ càng vì SQL Injection là kỹ thuật rất cơ bản.
- Ngoài ra, có thông tin rằng cấu hình không đúng hoặc máy chủ thử nghiệm đã được triển khai mà thiếu kiểm soát bảo mật đủ mạnh. Ví dụ: Bkav sau đó xác nhận rằng hơn 200 khách hàng bị lộ thông tin do "sai cấu hình" một hệ thống thử nghiệm.

3.3. Hậu quả

- Mã nguồn và thông tin nội bộ của Bkav bị rao bán với giá khoảng 290 000 USD hoặc hơn.
- Uy tín của Bkav – một công ty bảo mật – bị ảnh hưởng bởi việc để xảy ra lỗ hổng bảo mật cơ bản. (Tổn hại hình ảnh: "chuyên gia bảo mật mà lại bị lỗi cơ bản")
- Rò rỉ dữ liệu hơn 200 khách hàng của Bkav (họ tên, số điện thoại hoặc email) trên diễn đàn, do hệ thống thử nghiệm bị cấu hình sai.

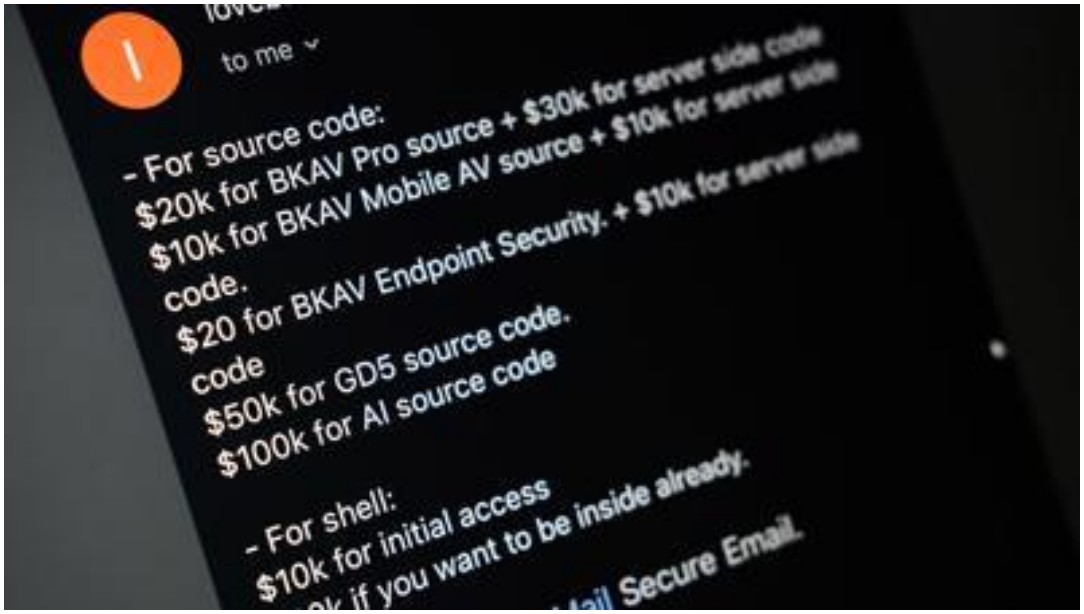

Figure 2. Bộ dữ liệu được thành viên seasalt123 chia sẻ trên một website chuyên mua bán dữ liệu (cre: https://mst.gov.vn/hon-200-khach-hang-bkav-bi-lo-thong-tin-197154354.htm)

4. Kết Luận

SQL Injection là lỗ hổng cổ điển nhưng luôn nguy hiểm vì cho phép kẻ tấn công can thiệp trực tiếp vào cơ sở dữ liệu, gây rò rỉ, sửa/xóa dữ liệu và leo thang đặc quyền. Thực tế cho thấy chỉ một điểm yếu "nhỏ" cũng có thể kéo theo thiệt hại lớn về dữ liệu, vận hành và uy tín. Để phòng chống hiệu quả ta cần thực hiện nhiều lớp bảo mật: (1) từ gốc mã nguồn bằng Prepared Statements/Parameterized Queries và Stored Procedures an toàn; (2) kiểm soát đầu vào (validation, sanitization) và nguyên tắc quyền tối thiểu trong cơ sở dữ liệu; (3) lớp bảo vệ bên ngoài với WAF và cấu hình hệ thống chuẩn; (4) giám sát liên tục bằng log, SIEM, quét tự động kết hợp review mã nguồn thủ công.