1. **Tình hình nghiên cứu trên thế giới?**

Tình hình nghiên cứu về an ninh cơ sở dữ liệu và ứng phó với các mối đe dọa liên quan đang được quan tâm rộng rãi trên toàn thế giới. Các nhà nghiên cứu và các tổ chức an ninh thông tin đều đang tập trung vào việc tìm ra các phương pháp hiệu quả để bảo vệ cơ sở dữ liệu khỏi các cuộc tấn công như SQL Injection, Cross-Site Scripting, và việc nghe lén thông tin đăng nhập.

Một số xu hướng và tiến triển trong lĩnh vực này bao gồm:

1. Phát triển các công cụ phát hiện và ngăn chặn tấn công: Các nhà nghiên cứu đang tạo ra các công cụ tự động để phát hiện và ngăn chặn các mối đe dọa an ninh cơ sở dữ liệu. Các công cụ này thường kết hợp cả kỹ thuật tĩnh và động để phát hiện lỗ hổng và tấn công.
2. Nghiên cứu về mã hóa dữ liệu: Mã hóa dữ liệu đang trở thành một phương tiện quan trọng để bảo vệ thông tin trước khi được lưu trữ trong cơ sở dữ liệu. Các phương pháp mã hóa tiên tiến như mã hóa dựa trên khóa công khai, mã hóa đối xứng và mã hóa băm đang được nghiên cứu và phát triển để cung cấp tính an toàn cao.
3. Phát triển các quy trình và phương pháp thử nghiệm an ninh: Các nhà nghiên cứu đang tạo ra các kỹ thuật thử nghiệm mới để đánh giá tính an toàn của cơ sở dữ liệu và ứng dụng. Điều này bao gồm cả việc phát triển các kịch bản tấn công mô phỏng để kiểm tra tính bảo mật của các hệ thống.
4. Nghiên cứu về phản ứng và phục hồi sau tấn công: Mặc dù việc ngăn chặn các cuộc tấn công là quan trọng, nhưng cũng quan trọng là có các biện pháp phản ứng và phục hồi sau tấn công hiệu quả. Các nghiên cứu đang tập trung vào việc phát triển các phương pháp và công cụ để phục hồi dữ liệu và hệ thống sau khi bị tấn công.

Tóm lại, tình hình nghiên cứu về an ninh cơ sở dữ liệu trên thế giới đang đặc biệt sôi động và đa dạng, với sự quan tâm ngày càng tăng về việc bảo vệ thông tin quan trọng khỏi các cuộc tấn công và lỗ hổng an ninh.

1. **Bài báo này có điểm gì khác với các hướng trên?**

Bài báo này đưa ra một phương pháp cụ thể để giải quyết vấn đề an ninh cơ sở dữ liệu bằng cách tạo ra một mô hình cụ thể cho cơ sở dữ liệu và ứng dụng liên quan. Phương pháp này tập trung vào việc tự bảo vệ cơ sở dữ liệu tại cấp độ của cơ sở dữ liệu chính mà không phụ thuộc vào các biện pháp an ninh ở các mức độ khác như mạng, hệ điều hành hoặc ứng dụng khách. Một số điểm khác biệt chính có thể được tóm tắt như sau:

1. Hướng tiếp cận tập trung vào mô hình cơ sở dữ liệu: Bài báo đề xuất một mô hình cụ thể cho cơ sở dữ liệu và ứng dụng liên quan, trong đó các quyền và hạn chế được thiết lập trực tiếp tại cấp độ của cơ sở dữ liệu, thay vì thông qua bảng người dùng như cách thường thấy. Điều này tạo ra một cách tiếp cận tự bảo vệ đặc biệt cho cơ sở dữ liệu.
2. Sử dụng các stored procedures và kiểm tra dữ liệu: Bài báo đề xuất việc sử dụng stored procedures để thực hiện tất cả các hoạt động trên cơ sở dữ liệu, đồng thời kiểm tra và xác thực dữ liệu đầu vào. Điều này giúp ngăn chặn các cuộc tấn công như SQL Injection bằng cách ngăn chặn các câu lệnh SQL động và kiểm tra tính hợp lệ của dữ liệu đầu vào.
3. Cân nhắc về việc sử dụng SSL cho việc truyền tải dữ liệu: Bài báo nhấn mạnh về việc sử dụng SSL để bảo vệ dữ liệu được truyền từ máy khách đến máy chủ, đặc biệt là trong quá trình xác thực người dùng. Điều này giúp ngăn chặn việc nghe lén và đánh cắp thông tin đăng nhập.
4. Phương pháp kiểm thử và thử nghiệm: Bài báo đề xuất các phương pháp kiểm thử an ninh cụ thể, bao gồm kiểm thử mã nguồn và kiểm thử đen, để đảm bảo tính an toàn của cơ sở dữ liệu và ứng dụng.

Tóm lại, bài báo này đi sâu vào việc xem xét và giải quyết vấn đề an ninh cơ sở dữ liệu thông qua việc phát triển một mô hình cụ thể và các biện pháp an ninh đặc biệt tại cấp độ của cơ sở dữ liệu chính.

1. **Điểm mạnh & điểm yếu là gì?**

Điểm mạnh:

1. Phương pháp cụ thể và chi tiết: Bài báo cung cấp một phương pháp cụ thể để giải quyết vấn đề an ninh cơ sở dữ liệu, bao gồm cả việc thiết kế một mô hình cơ sở dữ liệu đặc biệt và các biện pháp bảo vệ tại cấp độ cơ sở dữ liệu chính.
2. Sử dụng stored procedures và kiểm tra dữ liệu đầu vào: Bài báo tạo ra một cơ chế tự bảo vệ cho cơ sở dữ liệu, giúp ngăn chặn các cuộc tấn công như SQL Injection và Cross-Site Scripting.
3. Đề xuất các phương pháp kiểm thử an ninh: Bài báo đề xuất các phương pháp kiểm thử cụ thể để đảm bảo tính an toàn của cơ sở dữ liệu và ứng dụng.

Điểm yếu:

1. Phụ thuộc vào mô hình cụ thể: Phương pháp đề xuất trong bài báo có thể phụ thuộc nhiều vào mô hình cụ thể được đề xuất, điều này có thể làm giảm tính linh hoạt và khả năng mở rộng của phương pháp trong các môi trường khác nhau.
2. Phương pháp kiểm thử chưa được cụ thể hóa: Mặc dù bài báo đề xuất các phương pháp kiểm thử an ninh, nhưng không cung cấp chi tiết cụ thể về cách triển khai và thực hiện các phương pháp này trong thực tế. Điều này có thể làm cho việc thực hiện trở nên khó khăn đối với các nhà phát triển và kiểm thử viên.
3. **Bài báo này nói về cái gì, vấn đề gì?**

Bài báo tập trung vào việc đề xuất một phương pháp kiểm thử an ninh cho phần mềm ứng dụng cơ sở dữ liệu, nhấn mạnh rằng mặc dù các nhà phát triển thường tập trung vào đáp ứng các yêu cầu chức năng, nhưng an ninh cũng cần được coi trọng từ đầu. Nó đề xuất các kỹ thuật và phương tiện bảo mật như kiểm tra mã nguồn tĩnh và động, mã hóa dữ liệu và kiểm thử tự động để phát hiện và ngăn chặn các lỗ hổng an ninh như SQL Injection và Cross-Site Scripting (XSS). Đồng thời, bài báo cũng đề xuất một mô hình cơ sở dữ liệu cụ thể với các đặc điểm như hạn chế quyền truy cập trực tiếp vào bảng cơ sở dữ liệu và sử dụng thủ tục lưu trữ để thực hiện logic kinh doanh. Cuối cùng, nó nhấn mạnh về sự quan trọng của việc kiểm tra an ninh một cách có hệ thống và liên tục, và rằng an ninh phần mềm là một quá trình liên tục cần được chú ý.Top of Form

1. **Cách giải quyết ra sao?**

Cách giải quyết được đề xuất trong bài báo là sử dụng một số kỹ thuật và phương pháp bảo mật cụ thể:

1. **Kiểm tra mã nguồn tĩnh và động**: Đề xuất sử dụng phương pháp kiểm tra mã nguồn của ứng dụng để phát hiện các lỗ hổng an ninh từ các điểm yếu trong mã nguồn. Kiểm tra động có thể thực hiện bằng cách chạy các công cụ tự động để phát hiện lỗ hổng và các điểm yếu trong quá trình chạy của ứng dụng.
2. **Mã hóa dữ liệu**: Đề xuất sử dụng kỹ thuật mã hóa để bảo vệ dữ liệu khỏi việc truy cập trái phép. Mã hóa dữ liệu giúp ngăn chặn việc đọc thông tin nhạy cảm từ các bên không có quyền truy cập.
3. **Kiểm thử tự động**: Sử dụng các công cụ kiểm thử tự động để phát hiện các lỗ hổng phổ biến như SQL Injection và Cross-Site Scripting (XSS). Các công cụ này có thể tự động thực hiện các kịch bản kiểm thử để kiểm tra tính bảo mật của ứng dụng.
4. **Hạn chế quyền truy cập trực tiếp vào cơ sở dữ liệu**: Đề xuất hạn chế quyền truy cập trực tiếp vào cơ sở dữ liệu thông qua việc sử dụng thủ tục lưu trữ. Thay vì cho phép người dùng truy cập trực tiếp vào bảng cơ sở dữ liệu, thủ tục lưu trữ sẽ thực hiện các thao tác trên dữ liệu thông qua giao diện đã được kiểm soát.
5. **Sử dụng giao thức mã hóa Secure Sockets Layer (SSL)**: Đề xuất sử dụng SSL để bảo vệ việc truyền tải thông tin đăng nhập từ người dùng đến máy chủ. SSL sẽ mã hóa thông tin đăng nhập để ngăn chặn việc đánh cắp thông tin trên đường truyền.

Tổng thể, các đề xuất này tạo ra một cấu trúc bảo mật toàn diện cho phần mềm ứng dụng cơ sở dữ liệu, từ việc kiểm tra mã nguồn và kiểm thử tự động đến việc áp dụng các biện pháp bảo mật cụ thể để bảo vệ dữ liệu và ngăn chặn các cuộc tấn công từ các hacker.

1. **Giải thích mô hình or thuật toán?**

Mô hình được đề xuất trong bài báo tập trung vào việc bảo vệ cơ sở dữ liệu bằng cách giảm thiểu quyền truy cập trực tiếp vào bảng cơ sở dữ liệu và thực hiện tất cả các thao tác thông qua các thủ tục lưu trữ. Điều này giúp tạo ra một lớp trừu tượng giữa ứng dụng và cơ sở dữ liệu, giảm thiểu nguy cơ của các lỗ hổng an ninh thông qua việc thực hiện logic kinh doanh và kiểm tra dữ liệu tại mức độ cơ sở dữ liệu.

Thuật toán chính trong mô hình này là thiết kế và triển khai các thủ tục lưu trữ với các quy tắc và logic cụ thể để kiểm tra quyền truy cập của người dùng, xác thực dữ liệu và thực hiện các thao tác dữ liệu an toàn. Cụ thể, mỗi thủ tục lưu trữ được thiết kế với các điều kiện kiểm tra quyền truy cập, kiểm tra tính hợp lệ của dữ liệu đầu vào và thực hiện các thao tác dữ liệu chỉ sau khi các điều kiện này được kiểm tra.

1. **Em đánh giá gì về bài báo này?**

Bài báo này cung cấp một cái nhìn tổng quan về vấn đề quan trọng của an ninh cơ sở dữ liệu và cách thức thử nghiệm an ninh để giảm thiểu các lỗ hổng trong hệ thống phần mềm. Điểm mạnh của bài báo là nó trình bày một phương pháp thử nghiệm an ninh cụ thể, bao gồm xem xét mã nguồn và đánh giá các lỗ hổng, cùng với việc đề xuất các biện pháp cụ thể để bảo vệ cơ sở dữ liệu.

Tuy nhiên, bài báo có thể trở nên khá kỹ thuật và phức tạp đối với những người không có kiến thức chuyên sâu về an ninh thông tin hoặc phát triển phần mềm. Đôi khi, nó có thể đòi hỏi đọc giả có một mức độ kiến thức cơ bản về lập trình và quản trị cơ sở dữ liệu để hiểu hết nội dung.

Ngoài ra, bài báo có thể cung cấp thêm giá trị nếu đi sâu vào các ví dụ cụ thể hoặc thực tiễn áp dụng, giúp người đọc hiểu rõ hơn về cách thức thực hiện các biện pháp an ninh trong thực tế. Điều này có thể giúp đọc giả áp dụng những kiến thức từ bài báo vào công việc của họ một cách dễ dàng hơn.

1. **Bài báo đề xuất cái gì?**

Bài báo đề xuất một phương pháp và mô hình cụ thể để bảo vệ an ninh của cơ sở dữ liệu. Phương pháp này bao gồm việc sử dụng stored procedures và kiểm tra dữ liệu đầu vào để tạo ra một cơ chế tự bảo vệ cho cơ sở dữ liệu, ngăn chặn các cuộc tấn công như SQL Injection và Cross-Site Scripting. Đồng thời, bài báo cũng đề xuất các phương pháp kiểm thử an ninh cụ thể để đảm bảo tính an toàn của cơ sở dữ liệu và ứng dụng.

1. **Điểm mạnh và điểm yếu của đề xuất này ntn?**

Điểm mạnh:

1. **Bảo vệ tại cấp độ cơ sở dữ liệu:** Phương pháp đề xuất tập trung vào việc bảo vệ an ninh tại cấp độ cơ sở dữ liệu, thay vì phụ thuộc hoàn toàn vào các biện pháp bảo vệ tại mức độ ứng dụng hay hệ thống.
2. **Sử dụng stored procedures và kiểm tra dữ liệu đầu vào:** Sử dụng stored procedures giúp hạn chế việc thực thi truy vấn SQL trực tiếp từ ứng dụng, từ đó ngăn chặn các cuộc tấn công SQL Injection. Kiểm tra dữ liệu đầu vào cũng giúp loại bỏ hoặc giảm thiểu nguy cơ của các cuộc tấn công Cross-Site Scripting.
3. **Thiết kế cơ sở dữ liệu chặt chẽ:** Đề xuất yêu cầu thiết kế cơ sở dữ liệu theo một cách cụ thể và an toàn, với việc sử dụng các stored procedures để thực hiện các thao tác trên dữ liệu. Điều này giúp cải thiện tính an toàn và bảo mật của hệ thống.

Điểm yếu:

1. **Phụ thuộc vào mô hình cụ thể:** Phương pháp đề xuất có thể phụ thuộc nhiều vào mô hình cụ thể được đề xuất, điều này có thể làm giảm tính linh hoạt và khả năng mở rộng của phương pháp trong các môi trường khác nhau.
2. **Khó khăn trong triển khai và thực hiện:** Việc triển khai và thực hiện phương pháp có thể đòi hỏi sự hiểu biết sâu rộng về cả cơ sở dữ liệu và ứng dụng, cũng như kỹ năng kỹ thuật cao từ các nhà phát triển và quản trị hệ thống. Điều này có thể làm tăng chi phí và thời gian triển khai của dự án.
3. **Các mối đe dọa được đề cập trong bài báo này là gì? Chỗ nào**
4. SQL Injection Attacks: Các cuộc tấn công này xảy ra khi ứng dụng sử dụng dữ liệu nhập từ người dùng để tạo các câu lệnh SQL mà không được kiểm tra đúng cách. Điều này có thể dẫn đến việc thực thi các câu lệnh SQL độc hại mà người dùng có thể nhập vào.
5. Cross-Site Scripting (XSS) Attacks: Loại tấn công này cho phép kẻ tấn công chèn mã script vào các trang web mà người dùng khác có thể truy cập, làm cho mã script độc hại được thực thi trên máy tính của họ.
6. Eavesdropping và Password Theft: Các kẻ tấn công có thể nghe lén lưu lượng mạng giữa máy chủ và máy khách để đánh cắp thông tin đăng nhập của người dùng, như tên người dùng và mật khẩu.

Những mối đe dọa này đều là các vấn đề lớn trong an ninh cơ sở dữ liệu và đề xuất giải pháp để ngăn chặn chúng, bao gồm việc sử dụng các thủ tục lưu trữ để kiểm tra và xác nhận dữ liệu nhập và việc sử dụng mã hóa cho việc truyền tải thông tin nhạy cảm như mật khẩu.

(Trích đoạn từ bài báo:

"3.1 SQL Injection attacks SQL Injection vulnerability results from the application’s use of user input in constructing database statements. If user input will be assigned to variables input1 and input2 and if the contents of that variable will be used to dynamically create SQL statement query like:

query = “SELECT UserId FROM Users WHERE uname =’” +input1+”’ and passw= ’”+input2+”’”

it is obvious that if input1=admin, and input2=trust, query will be:

query = “SELECT UserId FROM Users WHERE uname=’admin’ and passw=’trust’”

and query will be sent to database to be executed. If user is allowed to put anything to appropriate web form field (our variable input), he/she might try to put the following [15]:

1. input1=anyuser, input2=pass’ or ‘1’=’1
2. input1= anyuser' or 1=1 LIMIT 1;#, input2=pass
3. input1=anyuser, input2=pass’ AND 1=0 UNION SELECT (case+when+(USER()='root@localhost')+then+ 2+else+1+end) AND ‘1’=’1"

Đây là phần trong bài báo mô tả về các cuộc tấn công SQL Injection và mô tả cách chúng có thể được thực hiện.)Top of Form

1. **Bài báo này giải quyết vấn đề gì? Chỗ nào? Thông qua mô hình hay thuật toán nào?**

Bài báo này giải quyết vấn đề về an ninh cơ sở dữ liệu, nhằm bảo vệ cơ sở dữ liệu khỏi các mối đe dọa như SQL Injection, Cross-Site Scripting, và việc nghe lén thông tin đăng nhập.

Chỗ giải quyết vấn đề nằm ở phần "Proposal for Problem Solution" của bài báo, trong đó bài báo đề xuất một mô hình cụ thể để bảo vệ cơ sở dữ liệu. Mô hình này bao gồm các biện pháp bảo mật như:

1. Xây dựng quy trình phân quyền và kiểm soát quyền truy cập: Người dùng không được phép truy cập trực tiếp vào các bảng trong cơ sở dữ liệu, mà thay vào đó họ phải sử dụng các thủ tục lưu trữ có sẵn để thực hiện các thao tác như truy vấn, chèn, cập nhật hoặc xóa dữ liệu.
2. Xây dựng các thủ tục lưu trữ an toàn: Các thủ tục lưu trữ được thiết kế với logic tích hợp để kiểm tra quyền hạn của người dùng và xác thực dữ liệu đầu vào trước khi thực thi thao tác.
3. Sử dụng mã hóa: Dữ liệu được mã hóa để bảo vệ khỏi việc truy cập trái phép và lắng nghe thông tin đăng nhập.
4. Sử dụng SSL để bảo vệ truyền dữ liệu: Thông tin đăng nhập được truyền tải giữa máy chủ và máy khách bằng cách sử dụng SSL để ngăn chặn việc nghe lén.

Bài báo cũng đề xuất một số phương pháp thử nghiệm an ninh để đảm bảo tính hiệu quả của mô hình, bao gồm việc phân tích mã nguồn và thử nghiệm tự động các lỗ hổng an ninh.

(Trích đoạn từ bài báo:

"4 Proposal for Problem Solution As it was mentioned, the security of database cannot rely on security of other levels of software system. Database must be as much self-protected as possible. We have created the database with the following specifications:

1. users that have permissions and privileges to access to database application are not controlled through User table, but they are registered as regular database users without any administrative roles,
2. there is no one regular user (except the db owner) that can have any (select, insert, update, delete) permission on any table,
3. the only way of viewing, inserting, modifying or deleting data is through the stored procedures, where all users have permission of executing procedures,
4. in stored procedures there is no dynamically created queries that can be executed through execute\_sql(string) commands; queries are built with stored procedures parameters (see Figure 1)
5. stored procedures have built-in logic for checking the user rights and permissions on different business functionalities (through the tables with encrypted attributes),
6. stored procedures have built-in logic for checking the properties of parameters sent to them (width, black list words, black list encoded characters, etc.),
7. calls to stored procedures with suspicious parameters’ values are logged to special tables,
8. local database administrators may change user rights and permissions only through stored procedures that identify if request comes from LAN.")