BÁO CÁO BÀI TẬP

**Môn học: An toàn mạng**

**Tên chủ đề: Nộp bài tập buổi 2**

*GVHD: Nghi Hoàng Khoa*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT140.O11.ANTT

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Thị Hồng Lam | 20521518 | 20521518@gm.uit.edu.vn |
| 2 | Nguyễn Triệu Thiên Bảo | 21520155 | 21520155@gm.uit.edu.vn |
| 3 | Trần Lê Minh Ngọc | 21521195 | [21521195@gm.uit.edu.vn](mailto:21521195@gm.uit.edu.vn) |
| 4 | Huỳnh Minh Khuê | 21522240 | 21522240@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Yêu cầu 1 | 100% |
| 2 | Yêu cầu 2 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. Tóm tắt tất cả các Nguyên tắc Thiết kế Bảo mật (Security Design Principles) và đưa ra ví dụ cho từng nguyên tắc.

* Economy of mechanism
* Fail-safe defaults
* Complete mediation
* Open design
* Separation of privilege
* Least privilege
* Least common mechanism
* Psychological acceptability
* Isolation
* Encapsulation
* Modularity
* Layering
* Least astonishment

*Trả lời:*

* **Nguyên lý thiết kế tiết kiệm và đơn giản (Economy of Mechanism):** Nguyên tắc bảo mật cơ bản này xác định rằng các biện pháp bảo mật được triển khai trong phần mềm và phần cứng phải đơn giản và nhỏ gọn nhất có thể. Điều này sẽ giúp người kiểm tra dễ dàng kiểm tra các biện pháp bảo mật một cách kỹ lưỡng. Khi thiết kế bảo mật đơn giản, việc cập nhật hoặc sửa đổi thiết kế sẽ dễ dàng. Với một thiết kế phức tạp, đối thủ sẽ có nhiều cơ hội hơn để phát hiện ra những lỗ hổng. Cơ chế càng phức tạp thì càng có nhiều khả năng có thể sai sót có thể khai khác được. Nhưng khi đi vào thực tiễn, đây có lẽ là nguyên tắc khó thực hiện nhất. Bởi vì nhu cầu liên tục bổ sung các tính năng bảo mật ở cả phần cứng cũng như phần mềm, làm phức tạp thêm nhiệm vụ thiết kế và bảo mật.

*VD*: khi sử dụng các thuật toán mã hóa trong hệ thống, nên thiết lập những thuật toán đã được kiểm tra và sử dụng nhiều như AES hay SHA256 vì những thuật toán này đã được kiểm tra và tối ưu nhất thay vì sử dụng những thuật toán tự viết.

* **Nguyên lý ngầm định đảm bảo hoạt động (Fail-safe defaults)**: phương pháp “Fail-safe defaults” giúp tăng tính an toàn và độ tin cậy của hệ thống truy cập và bảo vệ dữ liệu bằng cách thiết lập trạng thái mặc định là “thiếu quyền truy cập” để đảm bảo rằng mọi yêu cầu truy cập phải được xác nhận và có quyền hạn tương ứng trước khi được chấp nhận. Điều này giúp ngăn ngừa các cuộc tấn công và xâm nhập từ các bên không có quyền truy cập.

*VD:* Hầu hết các hệ thống truy cập tệp tin và dịch vụ được bảo vệ trong hệ thống client/server hoạt động theo phương pháp này.

* **Nguyên lý thiết kế mở (Open design):** có nghĩa là thiết kế của một cơ chế bảo mật nên được công khai thay vì bí mật. Điều đó giúp cho những thiết kế này dễ dàng được kiểm tra và đánh giá bởi đông đảo chuyên gia, các lỗ hổng và vấn đề bảo mật có thể được phát hiện và sửa chữa một cách nhanh chóng.

*VD:* Mặc dù các khóa mã hóa phải được bảo mật, các thuật toán mã hóa nên được công khai để được kiểm tra bởi các chuyên gia và người dùng có thể tin tưởng vào chúng. Điều này là triết lý đằng sau chương trình của Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) trong việc tiêu chuẩn hóa các thuật toán mã hóa và bảng băm.

* **Nguyên lý tối thiểu quyền ưu tiên (Least privilege):** nguyên tắc "least privilege" đòi hỏi mỗi quy trình và mỗi người dùng trên hệ thống nên sử dụng tập hợp quyền hạn nhỏ nhất cần thiết để thực hiện nhiệm vụ của mình để giảm rủi ro, ngăn chặn tiềm năng của tấn công, hạn chế thiệt hại do lỗi và quản lý rủi ro an toàn trong hệ thống.

*VD*: Chính sách bảo mật hệ thống có thể xác định và định nghĩa các vai trò khác nhau của người dùng hoặc một quy trình nào đó. Mỗi vai trò chỉ được gán các quyền hạn cần thiết để thực hiện chức năng của nó.Trừ khi được cấp phép rõ ràng, người dùng hoặc quy trình không nên có quyền truy cập vào tài nguyên được bảo vệ.

* **Nguyên lý cơ chế phổ biến nhất (Least common mechanism):** nguyên tắc thiết kế bảo mật giảm thiểu các chức năng được chia sẻ bởi những người dùng khác nhau, mang lại sự bảo mật lẫn nhau. Nguyên tắc này làm giảm số lượng đường dẫn truyền thông và do đó làm giảm hơn nữa việc triển khai phần cứng và phần mềm. Nguyên tắc này làm giảm nguy cơ truy cập không mong muốn vào hệ thống vì nó trở nên dễ dàng xác minh xem có một số quyền truy cập không mong muốn vào chức năng dùng chung hay không.

*VD*: nếu có nhu cầu được truy cập vào tệp hoặc ứng dụng bởi nhiều người dùng, thì những người dùng này nên sử dụng các kênh riêng biệt để truy cập các tài nguyên này, điều này giúp tránh những hậu quả không lường trước được và có thể gây ra sự cố bảo mật.

* **Nguyên lý cô lập (Isolation):** nguyên tắc “cô lập” được áp dụng trong ba ngữ cảnh. Đầu tiên, các hệ thống truy cập công cộng phải được cô lập khỏi các tài nguyên quan trọng (dữ liệu, quy trình, v.v.) để ngăn chặn việc tiết lộ hoặc giả mạo. Thứ hai, các quy trình và tệp của từng người dùng phải được cô lập với nhau ngoại trừ trường hợp được chia sẻ với mục đích rõ ràng.Và cuối cùng, các cơ chế bảo mật cần được cô lập nhằm ngăn chặn việc truy cập vào các cơ chế đó.

*VD:* Trong trường hợp độ nhạy cảm hoặc mức độ quan trọng của thông tin cao, các tổ chức có thể giới hạn số lượng hệ thống lưu trữ dữ liệu và cách ly chúng, về mặt vật lý hoặc logic. Cách ly vật lý có thể bao gồm việc đảm bảo rằng không có kết nối vật lý nào tồn tại giữa tài nguyên thông tin truy cập công khai của tổ chức và thông tin quan trọng của tổ chức.

* **Nguyên lý đóng gói (Encapsulation):** Nguyên tắc thiết kế bảo mật này là một dạng cô lập được thiết kế theo nguyên tắc hướng đối tượng. Ở đây các tiến trình của hệ thống được bảo vệ, chỉ có thể truy cập vào đối tượng dữ liệu của hệ thống và các tiến trình này chỉ có thể được gọi từ một điểm vào miền.

*VD:* nguyên lý đóng gói cũng có thể được áp dụng vào việc phân đoạn mạng bằng tường lửa. Các mạng riêng được tạo ra bằng cách phân cấp các VLANs và đặt tường lửa giữa chúng. Các phân đoạn riêng biệt được đóng gói trong vùng bảo mật của mình, sử dụng tường lửa làm ranh giới. Các tài liệu, máy chủ hay hệ thống nhạy cảm được đóng gói vào những phân đoạn riêng biệt, giúp ngăn chặn sự truy cập trái phép.

* **Nguyên lý lớp nguồn mở (Layering):** Nhiều lớp bảo mật phải được sử dụng để chống kẻ tấn công truy cập thông tin quan trọng. Việc áp dụng nhiều lớp bảo mật mang lại nhiều rào cản nếu kẻ tấn công cố gắng truy cập vào hệ thống được bảo vệ.

*VD*: Để bảo mật thông tin người dùng trong 1 hệ thống, cần kết hợp nhiều lớp bảo mật khác nhau như kiểm soát quyền truy cập (thiết lập mật khẩu, xác thực đa yếu tố,..), bảo mật ứng dụng (để chống các lỗi tấn công phổ biến như SQL injection), mã hóa dữ liệu (để bảo vệ dữ liệu khi truyền qua các kênh truyền không an toàn, chống đánh cắp dữ liệu).

* **Nguyên lý bất ngờ nhỏ nhất (Least astonishment):** nguyên tắc này liên quan tới việc thiết kế chương trình hoặc giao diện người dùng sao cho người dùng không cảm thấy khó hiểu, ít hiểu lầm tính năng hoặc tạo ra lỗ hổng bảo mật.

*VD*: như các tin nhắn báo lỗi của hệ thống (error messages), chúng cần phải báo cáo ngay lập tức khi có sự cố, và phải có giải thích rõ ràng sự cố là gì cùng với đề xuất giải pháp thích hợp.

* **Nguyên lý kiểm soát đầy đủ (complete mediation):** nguyên lý này yêu cầu tất cả các truy cập đến các đối tượng đều được kiểm tra để đảm bảo rằng người dùng được phép truy cập. Chủ thể sử dụng phải có quyền truy cập mới được cấp tài nguyên để đọc một đối tượng hệ điều hành. Nếu chủ thể muốn đọc lại nhiều lần thì hệ thống phải kiểm tra xem chủ thể có được phép đọc nhiều lần hay không. Hệ thống thường lưu lại các kết quả của lần kiểm tra đầu tiên và lần truy cập thứ 2 sẽ dựa vào kết quả lưu trữ tạm này. Nguyên lý này tránh việc kiểm tra một lần nhưng đọc lại nhiều lần.

*VD:* dịch vụ tên miền DNS lưu trữ tạm thông tin ánh xạ các tên máy trạm thành địa chỉ IP. Nếu kẻ tấn công hủy hoại lưu trữ tạm bằng việc cấy bản ghi kết hợp các địa chỉ IP ma với một tên thì một máy trạm sẽ định tuyến các kết nối đến máy trạm ở xa một cách không chính xác.

* **Nguyên lý tách nhỏ quyền ưu tiên (separation of privilege):** nguyên lý này yêu cầu một hệ thống không nên cấp quyền dựa chỉ trên một điều kiện riêng lẻ. Chỉ khi có nhiều hơn một điều kiện thỏa mãn thì mới cấp quyền truy cập. Điều này cung cấp sự kiểm soát chi tiết hơn trên tài nguyên và cũng tăng thêm mức đảm bảo cho truy cập có thẩm quyền.

*VD:* Trên một số phiên bản dựa vào Berkeley của hệ điều hành UNIX, người dùng không được thay đổi tài khoản của họ thành tài khoản root, trừ khi bắt buộc phải đáp ứng được hai điều kiện: người dùng phải biết mật khẩu root và phải thuộc nhóm có GID 0. Nếu chỉ thỏa mãn 1 trong 2 thì cũng không được phép truy cập.

* **Nguyên lý thuận tiện truy cập (psychological acceptability):** nguyên lý này yêu cầu các cơ chế an toàn không nên làm cho tài nguyên khó truy cập hơn so với trường hợp không có các cơ chế an toàn. Cấu hình và thực hiện chương trình nên càng dễ dàng, càng trực quan càng tốt và mọi đầu ra phải rõ ràng, trực tiếp và có ích. Tuy nhiên trong thực tế, cơ chế an toàn có thể phải cộng thêm “tác dụng phụ” nào đó nhưng phải đảm bảo điều đó tối thiểu và hợp lý.

*VD:* Khi người dùng nhập mật khẩu sai khi đăng nhập thì hệ thống nên từ chối yêu cầu đăng nhập và ra thông báo đăng nhập thất bại. Nếu báo là sai mật khẩu thì trong trường hợp người dùng là attacker, họ sẽ biết tên tài khoản và chỉ việc dò mật khẩu để đăng nhập.

* **Nguyên lý mô-đun hóa (modularity):** nguyên lý này yêu cầu phát triển các chức năng dưới dạng module riêng biệt, được dùng để thiết kế và triển khai hệ thống, cơ chế. Các module này kết hợp lại tạo thành các chức năng và dịch vụ phổ biến, chẳng hạn như chức năng mã hóa/giải mã, thường các tính năng này sẽ gộp lại thành các module chung lớn hơn. Các module tuân thủ nguyên tắc này có khả năng chuyển đổi sang công nghệ mới hay nâng cấp tính năng mà không cần phải thiết kế lại toàn bộ hệ thống.

*VD:* Nhiều giao thức và ứng dụng sử dụng các chức năng mật mã. Thay vì triển khai các chức năng này trong từng giao thức hoặc ứng dụng riêng lẻ, ta có thể tạo ra một module mật mã chung để được gọi bởi nhiều giao thức và ứng dụng. Việc này giúp tập trung thiết kế và triển khai an toàn vào một module mật mã duy nhất, bao gồm các cơ chế bảo vệ module khỏi can thiệp.

2. Hãy xem xét mã nguồn sau đây để cho phép truy cập vào một tài nguyên:

DWORD dwRet = IsAccessAllowed(...);

if (dwRet == ERROR\_ACCESS\_DENIED) {

//Security check failed

//Inform user that access is denied

} else {

//Security check OK.

//Do something }

a. Giải thích lỗi bảo mật trong chương trình này.

b. Viết lại mã để tránh lỗ hổng.

*Trả lời:*

a. Chương trình có lỗi bảo mật ở phần điều kiện của if. Nếu kết quả của IsAccessAllowed là ERROR\_ACCESS\_DENIED thì truy cập bị từ chối, còn lại mặc định là cho phép truy cập. Tuy nhiên kết quả của hàm IsAccessAllowed không chỉ có ACCESS\_ACCEPTED mà còn có các kết quả khác (như NULL) nên trong trường hợp ra kết quả khác thì chương trình vẫn cho phép truy cập.

b. Ta sẽ sửa lại phần điều kiện if và code liên quan như sau:

DWORD dwRet = IsAccessAllowed(...);

if (dwRet == ACCESS\_ACCEPTED) {

//Security check OK.

//Do something

} else {

//Security check failed

//Inform user that access is denied

}

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)