|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HỌC VIỆN KỸ THUẬT MẬT MÃ  **KHOA CÔNG NGHỆ THÔNG TIN**  Logo HvKTMM  BÁO CÁO MÔN HỌC  **AN TOÀN HỆ ĐIỀU HÀNH**  **Đề tài:**  **TÌM HIỂU VỀ INTEGRITY FLAWS**   |  |  | | --- | --- | | **Sinh viên thực hiện:** | **Trần Cao Minh Bách – AT150204**  **Vũ Thị Ánh - AT150504**  **Bạch Trường An – AT150201**  **Phạm Đỗ Thuỳ Linh**  **Nguyễn Đình Hùng** | | **Giảng viên hướng dẫn:** | **Đồng Thị Thuỳ Linh** |   Hà Nội, 11-2021 |

**LỜI NÓI ĐẦU**

Tài liệu này được Khoa Công nghệ thông tin, học viện Kỹ thuật Mật mã soạn thảo và ban hành nhằm mục đích hướng dẫn sinh viên trình bày đồ án báo cáo thực tập cơ sở một cách khoa học và thống nhất. Bản thân tài liệu này được biên soạn phù hợp với các quy định về trình bày một đồ án tốt nghiệp, đồ án môn học. Vì vậy, sinh viên có thể sử dụng trực tiếp mẫu này như một template khi viết quyển đồ án của mình. Sinh viên cũng được khuyến khích tham khảo hoặc sử dụng mẫu này khi viết báo cáo thực tập, báo cáo kết quả nghiên cứu, và các đồ án môn học khác.

Chi tiết về nội dung của phần Lời nói đầu được trình bày trong Mục 1.3.3.

**LỜI CAM ĐOAN**

Lời cam đoan được viết tại đây. Xem cách viết lời cam đoan tại Mục 1.3.4.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ i](#_Toc88558453)

[DANH MỤC BẢNG BIỂU ii](#_Toc88558454)

[TÓM TẮT ĐỒ ÁN iii](#_Toc88558455)

[CHƯƠNG 1. KHÁI NIỆM CHUNG VỀ INTEGRITY FLAWS 1](#_Toc88558456)

[1.1 Tính toàn vẹn của hệ điều hành 1](#_Toc88558457)

[1.2 Các loại lỗ hổng toàn vẹn 1](#_Toc88558458)

[1.3 Đối tượng gây ảnh hưởng 2](#_Toc88558459)

[1.4 Đối tượng bị ảnh hưởng 3](#_Toc88558460)

[CHƯƠNG 2. PHƯƠNG THỨC VÀ ẢNH HƯỞNG GÂY RA TRÊN CÁC ĐỐI TƯỢNG 5](#_Toc88558461)

[2.1 Phương thức gây ra 5](#_Toc88558462)

[2.1.1 Đánh chặn (Interception) 5](#_Toc88558463)

[2.1.2 Các loại tài liệu tham khảo 5](#_Toc88558464)

[2.2 Trích dẫn tài liệu tham khảo 7](#_Toc88558465)

[2.3 Nộp đồ án 8](#_Toc88558466)

[2.3.1 Nộp bản cứng 8](#_Toc88558467)

[2.3.2 Nộp bản mềm 8](#_Toc88558468)

[2.4 Bản quyền kết quả nghiên cứu 8](#_Toc88558469)

[CHƯƠNG 3. CÁC HÌNH THỨC KHAI THÁC LỖ HỔNG 10](#_Toc88558470)

[3.1 Xác thực không đầy đủ 10](#_Toc88558471)

[3.2 Xác thực không đồng nhất 11](#_Toc88558472)

[3.3 Chia sẻ ngầm dữ liệu quan trọng 12](#_Toc88558473)

[3.4 Xác thực bất đồng bộ 13](#_Toc88558474)

[3.5 Nhận dạng/ Xác thực/ Uỷ quyền không đầy đủ 16](#_Toc88558475)

[3.6 Vi phạm giới hạn 17](#_Toc88558476)

[3.7 Lỗi logic 18](#_Toc88558477)

[KẾT LUẬN 20](#_Toc88558478)

[Kết luận chung 20](#_Toc88558479)

[Hướng phát triển 20](#_Toc88558480)

[Kiến nghị và đề xuất 20](#_Toc88558481)

[TÀI LIỆU THAM KHẢO 21](#_Toc88558482)

[PHỤ LỤC 22](#_Toc88558483)

[Phụ lục 1. Mẫu trang bìa chính của đồ án 22](#_Toc88558484)

[Phụ lục 2. Mẫu trang bìa phụ của đồ án 24](#_Toc88558485)

[Phụ lục 3. Mẫu nhận xét đồ án 26](#_Toc88558486)

# DANH MỤC HÌNH VẼ

[Hình 1.1 Sơ đồ khối của hệ thống 3](#_Toc9951164)

(Xem thêm Mục 1.3.7)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1 Kết quả thí nghiệm 3](#_Toc9951165)

(Xem thêm Mục 1.3.8)

# TÓM TẮT ĐỒ ÁN

Tóm tắt đồ án, có độ dài 1-2 trang, được trình bày tại đây. Chi tiết xem Mục 1.3.9.

# KHÁI NIỆM CHUNG VỀ INTEGRITY FLAWS

Nội dung chương đề cập tới khái niệm chung của intergrity flaws, nhằm đưa cái nhìn khách quan nhất về dạng lỗ hổng này.

## Tính toàn vẹn của hệ điều hành

Tính toàn vẹn của hệ điều hành liên quan đến việc đảm bảo rằng hệ điều hành hoạt động bình thường. Do đó, một lỗ hổng toàn vẹn của hệ điều hành cho phép người dùng (hoặc chương trình) làm cho hệ điều hành hoạt động không đáng chính xác và không an toàn. Tính toàn vẹn được xác định trước các vấn đề về độ tin cậy (gian lận và sai sót) và các vấn đề về bảo mật (kiểm tra tài nguyên và quyền riêng tư).

## Các loại lỗ hổng toàn vẹn

Tương ứng với mỗi thành phần của hệ thống thì việc chia lỗ hổng theo các thành phần này là cách đơn giản nhất để khái quát về nó. Với 6 loại sau:

1. Vật lý

Các lỗi liên quan đến vật lý thường là các lỗi mang tính chất khách quan như: thiên tai, thảm hoạ. Nhưng bên cạnh đó là các lỗi mang tính chủ quan như: việc bị đánh chặn viễn thông, độ bảo mật kém ở các thiết bị đầu cuối, truy cập không được phép vào phòng máy chứa các thông tin quan trọng.

1. Phần cứng

Là thành phần chính để người dùng có thể giao tiếp với các phần mềm thông qua cử chỉ vật lý. Vì vậy, những ảnh hưởng vật lý cơ bản như việc bị ngắt nguồn ở các thiết bị đầu cuối, hay các thiết bị đầu cuối không được đồng bộ tín hiệu đến với phần mềm hệ điều hành. Hoặc những lỗi như việc cung cấp các quyền một cách không chặt chẽ, khiến cho việc tiếp cận với các phần cứng dễ dàng, khiến cho nguy cơ bị phá hoại tăng cao.

1. Nhân viên – Nhân sự

Các lỗi liên quan đến nhân viên, nhân sự là các lỗi mang tính chủ quan là chính. Khi này những người có hành vi phá hoại và có ý định lạm dụng quyền lực để có thể thay đổi, đánh cắp hay phá hoại thông tin. Bên cạnh đó là khía cạnh chủ quan, khi nhân viên trong công ty bị dính bẫy của nhưng kẻ phá hoại từ đó lỡ khiếp cho kẻ xấu có cơ hội tiếp cận và phá hoại thông tin từ bên ngoài. Hay việc dán nhãn cho thông tin chưa chặt chẽ, người dùng trong hệ thống có thể phân quyền thiếu. Bên cạnh đó là vấn đề dính các loại Trojan Horse, worm hay virus cũng là những nguy cơ rất cao khiến mất tính toàn vẹn hệ thống xảy ra.

1. Thủ tục

Các thủ tục cài đặt không được kiểm soát chặt chẽ có thể dẫn tới việc người dùng cài đặt các phần mềm không rõ nguồn gốc, và có tồn tại các dạng virus, worn hay Trojan Horse, đe doạ đến sự toàn vẹn của hệ thống. Hoặc các thủ tục quá rườm rà, khiến người dùng không đọc kĩ cũng có thể khiến các cấu hình ban đầu không còn đảm bảo.

1. Phần mềm

Các phần mềm ứng dụng cho phép phân tách người dùng ra khỏi hệ thống một cách độc lập, sử dụng nó để có thể giao tiếp với hệ thống, nhằm giảm thiểu rủi ro đem lại cho hệ thống thực tế. Nhưng việc xây dựng và cấu hình hệ thống không chặt chẽ có thể khiến cho việc truy cập, kiểu soát các dữ liệu có thể xảy ra, cùng với đó là vấn đề logic trong chương trình, lỗi logic cũng có thể gây ra các ảnh hưởng ở nhiều mức độ khác nhau.

1. Hệ điều hành

Hệ điều hành cũng là một phần mềm, nên các vấn đề xảy ra ở trên phần mềm chính là lỗi trên hệ điều hành, nhưng việc tồn tại các lỗi ở hệ điều hành thì có thể gây ra các ảnh hưởng rất lớn, đặc biệt nghiêm trọng cho cả hệ thống. Vì vậy, việc đảm bảo hệ thống có một hệ điều hành ổn định, đảm bảo tính chuẩn xác và sự an toàn là vấn đề rất quan trọng.

## Đối tượng gây ảnh hưởng

Trong quá trình sử dụng, người dùng trong hệ thống là một thành phần có khả năng xâm phạm tính toàn vẹn của cài đặt thông qua lỗ hổng. Khi được đưa vào sử dụng, các lỗi tồn tại trên có thể cho phép truy cập trái phép vào tài nguyên không được phép thông qua những phương thức khác nhau, từ đó gây hại đến hệ thống.

Với những người dùng ứng dụng, họ là những người chưa được cấp các đặc quyền với hệ thống ngoài quyền có thể sử dụng các dịch vụ cơ bản mà hệ thống cho phép được sử dụng. Họ được chia thành người sử dụng và nhà sản xuất. Người sử dụng là người nhận được các uỷ quyền các thông tin sản phẩn từ một ứng dụng dựa trên máy tính. Nhà sản xuất là nơi phân phối và lập trình các ứng dụng đó, nhưng người thiết kế và triển khai các ứng dụng cụ thể tạo ra thông tin cho người tiêu dùng.

Việc người dùng gây ảnh hưởng tới hệ thống hay làm mất đi tính toàn vẹn của hệ thống là điều hết sức hiển nhiên. Vì lý do đó, việc nhà sản xuất phải thiết kế và phân tích làm sao đảm bảo được tính cô lập, nhằm giúp cho đảm bảo được sự an toàn cho chính hệ thống, cũng như những người dùng khác. Nhưng, bên cạnh đó, trong nhà sản xuất, các thông tin người dùng cũng là vấn đề gây ra nhiều tranh cãi, vì nhiều vụ việc nhân viên trong các nhà sản xuất tự ý xâm phạm, khiến cho tính chất bảo mật và toàn vẹn bị vi phạm nghiêm trọng, từ đó việc đảm bảo các chính sách ở lớp người dùng này là vấn đề quan trọng và cấp thiết.

Bên cạnh đó, nhóm sử dụng ở dạng dịch vụ được chia thành dnajg hệ thống và người phục vụ quản trị. Người bảo trì hệ thống là thành viên của đội ngũ nhân viên bảo trình bao gồm người vận hành, người lập trình hệ thống và kĩ sư quản lý chính. Những người chịu trách nhiệm bảo trì và tính sẵn có của hệ thống phải đảm bảo được tính sẵn và toàn vẹn để có thể cung cấp được dịch vụ một cách liên tục tại bất kì đâu. Nhưng cũng vì thế người vận hành có thể truy cập vào thông tin người dùng, thông tin hệ thống. Đôi khi việc lạm quyền có thể gây mất tính bảo mật và toàn vẹn cho hệ thống trên.

Và cuối cùng là hệ người dùng vãn lai, không có thông tin trên hệ thống. Đây là những người dùng không có đặc quyền truy cập vào các sản phẩm hay thiết bị, thay đổi hay sửa đổi thông tin. Nhưng việc cố gắng truy cập, khai thác sẽ là trái phép, và khả năng luôn luôn là việc tấn công và gây ra ảnh hưởng lớn cho toàn hệ thống nếu tồn tại các lỗ hổng có thể khai thác từ đó.

## Đối tượng bị ảnh hưởng

Các đối tượng bị ảnh hưởng là các đối tượng trực tiếp có mặt trong hệ thống như các tài nguyên của hệ thống. Khi bị ảnh hưởng, các tài nguyên sẽ bị ảnh hưởng theo hướng dây chuyền khiến cho ảnh hưởng có thể gây ra tác hại to lớn. Các tài nguyên có giá trị như là các thiết bị đầu cuối hay là thông tin của hệ thống, dịch vụ. Thông tin thì bao gồm tất các các tệp của hệ thống như: chương trình, dữ liệu, thư mục tệp của hệ thống và tất các thư mục của người dùng.

Dịch vụ thì đại diện cho các hoạt động của hệ thống cung cấp nơi người dùng cuối, thường luôn phải đảm bảo tính sẵn sàng, nhưng bên cạnh đó là đảm bảo tính toàn vẹn cho các thông tin liên quan đến dịch vụ đó. Nếu người dùng có quyền truy cập thì việc đảm bảo người dùng được sử dụng đúng các dịch vụ trên và không để người dùng lạm quyền, sử dụng trái phép các dịch vụ không được uỷ quyền.

Tài nguyên ở dạng thiết bị là các thành phần vật lý, có nhiều nguy cơ ảnh hưởng nhất, khi mà các thiết bị có thể bị các tác nhân khách quan gây ảnh hưởng, mỗi thiết bị trong hệ thống là một mắt xích quan trọng, vì thế việc đảm bảo tính toàn vẹn với thiết bị là rất cần thiết. Bên cạnh đó, phần cứng như các thiết bị không hoạt động, thì dịch vụ cũng không thể nào cung cấp cho các người dùng cuối được.

# PHƯƠNG THỨC VÀ ẢNH HƯỞNG GÂY RA TRÊN CÁC ĐỐI TƯỢNG

Loại lỗ hổng này xảy ra trên tất cả các thành phần của hệ thống thông tin. Chỉ cần sơ xuất trong quá trình cấu hình, cài đặt hệ thống, các đầu mối nhỏ trong cả một chuỗi cũng có thể gây ra lỗ hổng theo tính dây chuyền, khiến việc ngăn chặn lỗi hết sức muộn màng và gây nên nhiều tác hại nặng nề. Chương 2 sẽ đề cập tới các đối tượng đặc trưng có thể bị ảnh hưởng trong hệ thống. Bên cạnh đó là các ảnh hưởng này sẽ gây ra các tác động gì trên các đối tượng này.

## Phương thức gây ra

### Đánh chặn (Interception)

Đánh chặn là việc khiến các dịch vụ được cung cấp bị gián đoạn hoặc mất kết nối.

### Các loại tài liệu tham khảo

Các nguồn tài liệu tham khảo chính là sách, bài báo trong các tạp chí, bài báo trong các hội nghị khoa học, và các tài liệu tham khảo khác trên Internet. Sau đây là các quy định về cách liệt kê thông tin của các loại tài liệu tham khảo này. Các thông tin này phải đầy đủ để giúp cho người đọc có thể dễ dàng tìm lại được các tài liệu gốc khi cần. Chú ý cách trình bày các tài liệu tham khảo phải nhất quán và theo một format chung. Ví dụ: nếu tên các tác giả có phần tên riêng được viết tắt thì không nên viết đầy đủ tên riêng của bất cứ một tác giả nào, trong tài liệu đó và trong các tài liệu khác.

#### Sách

Thông tin về sách được sắp xếp theo thứ tự sau: tên các tác giả, *tên sách* (chữ nghiêng). Tên nhà xuất bản, và năm xuất bản. Ví dụ:

|  |
| --- |
| T. H. Cormen, C. E. Leiserson, and R. L. Rivet, *Introduction to Algorithm*. MIT Press, McGraw-Hill, 1990. |

#### Bài báo hoặc chương trong sách

Thông tin được sắp xếp theo thứ tự sau: tên các tác giả bài báo, “tên bài báo hoặc chương,” *tên sách* (chữ nghiêng), tên người soạn sách. Tên nhà xuất bản, năm xuất bản, số thứ tự trang của bài báo. Ví dụ:

|  |
| --- |
| J. W. DuBois, S. Schuetze-Coburn, S. Cumming, and D. Paolino, “Outline of discourse transcription,” in *Talking Data: Transcription and Coding in Discourse Research*, J. A. Edwards and M. D. Lampert, Ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, pp. 45-89.  G. O. Young, “Synthetic structure of industrial plastics,” in *Plastics*, vol. 3, *Polymers of Hexadromicon*, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64. |

#### Bài báo trong các tạp chí

Các thông tin về bài báo trong các tạp chí được sắp xếp theo thứ tự sau: tên tác giả, “tên bài báo,” *tên tạp chí* (chữ nghiêng), volume, số, số thứ tự trang của bài báo trong tạp chí, tháng và năm xuất bản. Ví dụ:

|  |
| --- |
| J. M. Airey, J. H. Rohfl, F. Brooks Jr., “Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments,” *Comptuer Graphics*, Vol. 24, No. 2, pp. 41-50, 1990.  R. E. Kalman, “New results in linear filtering and prediction theory,” *Journal of Basic Engineering*, ser. D, vol. 83, pp. 95-108, Mar. 1961.  E. H. Miller, “A note on reflector arrays,” *IEEE Transactions on Antennas and Propagation*, to be published. |

#### Bài báo trong các hội nghị có xuất bản

Các thông tin về bài báo trong các hội nghị có xuất bản kỷ yếu được sắp xếp theo thứ tự sau: tên tác giả, “tên bài báo,” *tên hội nghị* (chữ nghiêng), địa điểm và thời gian tổ chức hội nghị, số thứ tự trang trong kỷ yếu của hội nghị. Ví dụ:

|  |
| --- |
| S. Brandt, G. Nutt, T. Berk, M. Humphrey, “Soft Real time Application Execution with Dynamic Quality of Service Assurance,” in *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service*, Hawaii, USA, May 1998, pp. 154-163. |

#### Bài báo trong các hội nghị không xuất bản

Các thông tin về bài báo trong các hội nghị không xuất bản kỷ yếu được sắp xếp theo thứ tự sau: tên tác giả, “tên bài báo,” tên hội nghị, địa điểm và thời gian tổ chức hội nghị. Ví dụ:

|  |
| --- |
| K. Riley, “Language theory: Applications versus practice,” presented at the Conf. of the Modern Language Association, Boston, MA, December 27-30, 1990. |

#### Tài liệu trực tuyến

* Sách và bài báo trực tuyến:

|  |
| --- |
| J. Jones. (1991). *Networks* (2nd ed.) [Online]. Available: http://www.atm.com.  R. J. Vidmar. (1994). *On the use of atmospheric plasmas as electromagnetic reflectors* [Online]. Available FTP:atmnext.usc.edu Directory: pub/etext/1994 File: atmosplasma.txt |

* Nếu tham khảo tài liệu trên Internet, sinh viên phải ghi rõ URL của trang web và ghi ngày truy cập cuối cùng mà trang web đó vẫn còn có hiệu lực. Ví dụ:

|  |
| --- |
| http://www.dvb.org/, truy nhập cuối cùng ngày 10/1/2005. |

## Trích dẫn tài liệu tham khảo

Các nguyên tắc chung:

* Thông tin được trích dẫn nguyên văn phải được đặt trong dấu ngoặc kép.
* Số thứ tự của tài liệu được trích dẫn có thể được đặt ngay sau thông tin được trích dẫn hoặc đặt trước thông tin được trích dẫn. Ví dụ: Trong một bài viết có đoạn thông tin như sau:

|  |
| --- |
| Dân số Việt nam cho đến năm 2010 sẽ là 90 triệu người [15]. |

Cách viết như trên cho thấy, số liệu về dân số đã được tác giả tham khảo ở tài liệu tham khảo số 15 (báo cáo của uỷ ban dân số quốc gia). Cũng có thể trích dẫn thông tin trên như sau:

|  |
| --- |
| Theo báo cáo của uỷ ban dân số quốc gia [15], dân số Việt nam cho đến năm 2010 sẽ là 90 triệu người. |

* Một hình vẽ nếu được copy 100% thì phải ghi rõ nguồn gốc của hình vẽ đó ở phần chú thích của hình. Ví dụ:

|  |
| --- |
| Hình 3.1 Biểu đồ dân số các vùng của Việt nam [15] |

## Nộp đồ án

### Nộp bản cứng

Sinh viên (hoặc nhóm sinh viên với tối đa 3 thành viên làm chung một đề tài) nộp 01 quyển đồ án TTCS tại văn phòng khoa CNTT trước ngày bảo vệ ít nhất 03 ngày. Mỗi quyển đồ án phải có các đặc điểm sau:

* Được in một mặt hoặc hai mặt nhằm tiết kiệm không gian lưu trữ.
* Được đóng bìa mềm và có bìa bóng kính. Lưu ý: không dùng bìa cứng.
* Quyển phải có chữ ký của sinh viên sau Lời cam đoan và của giảng viên hướng dẫn.

### Nộp bản mềm

Sinh viên nộp bản mềm đồ án TTCS theo Giảng viên hướng dẫn. Mỗi sinh viên hoặc nhóm sinh viên phải nộp đủ toàn bộ các nội dung được làm trong đồ án TTCS (quyển file mềm dạng .docx và .pdf; project bao gồm cả mã nguồn và hướng dẫn cài đặt). Tất cả đặt trong cùng 1 thư mục và được đặt tên theo: manhom\_tendetai\_tengvhd\_nam.rar. Sinh viên hoặc nhóm sinh viên nộp cho GVHD dưới dạng link tới file đã gửi lên cloud (google, microsoft, v.v.) hoặc copy cho GVHD.

## Bản quyền kết quả nghiên cứu

Đồ án TTCS của sinh viên thường thuộc một trong hai kiểu sau đây:

* Thực hiện một nhiệm vụ nào đó để hoàn thành quá trình học tập. Sinh viên tự tìm điều kiện làm việc để hoàn thành đồ án. Do đó, sinh viên có quyền khai thác và sử dụng đồ án của mình vào những việc khác. Đồng thời, **Khoa Công nghệ thông tin có toàn quyền sử dụng các kết quả của đồ án và cam kết chia sẻ các kết quả trong bản đồ án cho tất cả những ai quan tâm và có yêu cầu**.
* Thực hiện một đề tài, một nhiệm vụ trong ý đồ chiến lược, trong chương trình khoa học và lao động sản xuất của giảng viên hướng dẫn, của cơ sở đào tạo (trung tâm, viện nghiên cứu, các công ty, v.v.), của cơ sở đào tạo cung cấp các phương tiện và điều kiện làm việc khác (máy móc, sách vở, điện thoại, truy cập internet, máy in, dụng cụ thiết bị thí nghiệm, kinh phí, v.v.). **Trong trường hợp này, toàn bộ bản quyền của đồ án không thuộc về sinh viên**. Sinh viên chỉ là một người tham gia thực hiện nên không được tuỳ tiện sử dụng ở những nơi khác với mục đích khác. Khi sinh viên thực hiện một đề tài theo hình thức này, vai trò của giảng viên hướng dẫn là định hướng, cung cấp các tài liệu tham khảo (nếu có), tạo điều kiện cơ sở vật chất (chỗ ngồi, máy tính, máy in, vật tư tiêu hao, phương tiện thông tin liên lạc, internet v.v.) để hoàn thành công việc.

# CÁC HÌNH THỨC KHAI THÁC LỖ HỔNG

Ở chương này sẽ đề cập tới những lỗi bảo mật từ đó thấy được tầm quan trọng của việc đảm bảo sự toàn vẹn của hệ thống nói chung và sự toàn vẹn của hệ điều hành nói riêng. Từng mục cũng sẽ có những ví dụ cụ thể, giúp cho ta có cái nhìn khái quát nhất từng lỗi.

## Xác thực không đầy đủ

Ở mức độ hoàn thiện nhất định, một chương trình với những câu lệnh có đặc quyền này gồm các yêu cầu các dịch vụ từ bước thứ 2, với một tập các câu lệnh có đặc quyền khác. Việc bảo vệ tính toàn vẹn cho hệ thống đòi hỏi việc xác thực phải kĩ lưỡng. Đối với hầu hết các hệ điều hành hiện nay, ranh giới giúp bảo toàn tính toàn vẹn hệ thống là ranh giới giữa chương trình điều khiển với chương trình người dùng. Khi đó, chương trình người dùng sẽ có những hạn chế nhất định, giúp cho hệ thống an toàn hơn. Bên cạnh đó, việc tách biệt này khiến cho các thành phần của hệ thống hoạt động độc lập, không làm gián đoạn quá trình hoạt động.

Các chương trình người dùng muốn sử dụng thì hệ thống sẽ gọi các chương trình con và nạp vào nhiều tham số. Và để gọi được thì người dùng cần thông qua chương trình điều khiển, lúc ấy, các dịch vụ được yêu cầu mới có thể thực thi. Việc tạp ra sự tách biệt này giữa các chương trình nhằm hạn chế được việc người dùng nào đó cố ý hay vô tình làm ảnh hưởng đến hoạt động của chương trình điều khiển, kéo theo ảnh hưởng dây chuyền các dịch vụ đang cung cấp cho người dùng khác. Nếu cơ chế kiểm tra các tham số đầu vào này không được chặt chẽ, việc kẻ tấn công có thể lợi dụng và đánh lừa chương trình điều khiển thực hiện được một yêu cầu vượt quá quyền hạn, từ đó gây ra sự mất an toàn cho toàn hệ thống. Và các trường cần phải xem xét cũng như kiểm tra kĩ như:

* Các kiểu và định dạng dữ liệu.
* Số lượng và thứ tự
* Phạm vi của giá trị
* Quyền truy cập vào các vị trí lưu trữ
* Tính nhất quán giữ các tham số (vị trí lữu trữ, …)

Từ những tham số trên, lỗ hổng này có thể có kịch bản nguy hiểm xảy ra nếu người dùng thành công trong việc nạp tham số không hợp lệ vào chương trình điều khiển. Việc chấp nhận một tham số bao gồm một địa chỉ bên ngoài không gian bộ nhớ được cấp phát cho người dùng đó có thể khiến:

* Chương trình có thể lấy được dữ liệu trái phép cho người dùng đó.
* Một tập hợp các điều kiện có thể được tạo ra gây ra sự cố cho hệ thống.
* Có thể thực thi được dòng lệnh nhằm điều khiển hệ thống.

Với một kịch bản có sẵn như sau có thể khiến cho hệ thống ảnh hưởng nghiêm trọng:

* Một lệnh, khi được thực thi, sẽ chuyển quyền điều khiển đến một điểm xác định trước trong chương trình của người dùng được nạp vào một thanh ghi.
* Một lệnh hệ thống sau đó được thực hiện làm cho các thanh ghi được lưu bằng chương trình điều khiển trong Vùng lưu đăng ký.
* Sau khi trả lại quyền kiểm soát cho người dùng, một lệnh gọi hệ thống khác sẽ được thực hiện. Trong số các tham số cho lệnh gọi hệ thống này, là một con trỏ (địa chỉ) phải trỏ đến một vị trí trong chương trình điều khiển. Địa chỉ này sẽ được sử dụng để chuyển quyền kiểm soát sang quy trình dịch vụ chương trình khác thích hợp. Đương nhiên, địa chỉ được cung cấp là vị trí trong Vùng lưu đăng ký nơi chuyển trở lại chương trình của người dùng đã được thực hiện bởi lệnh gọi hệ thống trước đó.
* Trong khi thực hiện lệnh gọi hệ thống thứ hai, quyền điều khiển được trả lại ở trạng thái điều khiển/giám sát cho người dùng, cho phép người dùng kiểm soát hệ thống.

## Xác thực không đồng nhất

Việc có nhiều định nghĩa của cùng một cấu trúc trong một hệ điều hành, thì sẽ có khả năng xảy ra mâu thuẫn giữa các định nghĩa này, việc đó sẽ tạo ra một lỗ hổng bảo mật. Lỗi này vượt ra ngoài lỗi xác thực thông số không đầy đủ. Một trường hợp có thể tồn tại trong đó mỗi quy trình của một số chương trình điều khiển sẽ kiểm tra hoàn toàn các điều kiện mà nó cho là hợp lệ; tuy nhiên, nhiều bộ tiêu chí không hoàn toàn nhất quán với nhau.

Một ví dụ về loại lỗ hổng này:

Hệ điều hành duy trì các thư mục (ví dụ: danh mục) của các tệp dữ liệu được sử dụng bởi hệ thống và người dùng. Nội dung của các thư mục này thường được truy cập bởi nhiều chương trình. Mỗi chương trình này quy chuẩn riêng về cách tạo thành một điều kiện hợp lệ trong hệ thống tệp.

Hãy xem xét điều gì đó cơ bản như các ký tự trong các tham số đại diện cho (các) tên của người dùng được cấp quyền truy cập tệp. Quy trình tạo mục nhập chỉ mục-tệp-chính có thể chấp nhận một ký tự (chẳng hạn như một ô trống được nhúng) là hợp lệ trong một tên quyền cụ thể; trong khi tất cả các chương trình giao diện khác sửa đổi/xóa các mục nhập chính tệp chỉ mục giả định các khoảng trống sẽ không bao giờ hợp lệ và do đó không chấp nhận chúng. Trong các điều kiện như vậy, các quyền đối với tệp cụ thể có thể được tạo (chẳng hạn như quyền truy cập được chia sẻ vào tệp) mà sau đó không thể bị xóa.

## Chia sẻ ngầm dữ liệu quan trọng

Để đảm bảo tính toàn vẹn, hệ điều hành phải có khả năng cô lập từng người dùng với tất cả những người khác và với chương trình điều khiển. Sự cô lập này liên quan đến các luồng điều khiển và thông tin. Bất cứ khi nào việc cô lập các thông tin không hoàn thiện, hệ thống có thể cho phép người dùng có đặc quyền cao hơn truy cập thông tin có đặc quyền thấp hơn hoặc có thể cho phép một người dùng truy cập thông tin của người dùng khác theo mong muốn của người dùng đó.

Trong nhiều hệ điều hành, phần điều khiển/chương trình của hệ điều hành chia sẻ không gian bộ nhớ với các chương trình người dùng, dưới dạng không gian làm việc hoặc là nơi thuận tiện để đưa thông tin liên quan đến chương trình người dùng đó. Đây là một chính sách thiết kế có chủ ý nhằm tạo điều kiện cho người dùng cá nhân hoá cho các tài nguyên mà họ sử dụng. Nếu người dùng yêu cầu hoạt động tệp hoặc các loại tài nguyên hệ thống khác, hệ thống sẽ duy trì thông tin và không gian làm việc cho yêu cầu của người dùng trong một khu vực. Bởi vì không gian làm việc được chia sẻ, nhưng ở một chế độ không có sẵn cho người dùng, những người triển khai hệ điều hành thường bất cẩn đối với trạng thái mà không gian làm việc còn lại sau khi nhận được yêu cầu của người dùng.

Ví dụ: chương trình điều khiển có thể sử dụng không gian làm việc như vậy để đọc trong chỉ mục chính của tệp người dùng cùng với mật khẩu được liên kết của họ như một phần của tìm kiếm dữ liệu do một người dùng nhất định yêu cầu. Chức năng này là cần thiết để hệ thống xác định rằng yêu cầu được hình thành đúng cách và được cấp quyền cho người dùng thực hiện yêu cầu. Nếu chương trình điều khiển thấy rằng yêu cầu không đúng, nó sẽ trả lại quyền điều khiển cho chương trình người dùng bắt nguồn yêu cầu, với dấu hiệu của lỗi trong yêu cầu. Tuy nhiên, trong ví dụ này, chương trình điều khiển không làm gì với thông tin còn lại trong không gian làm việc được chia sẻ. Do đó, người dùng giờ đây có thể truy cập không gian làm việc và lấy từ đó các mã định danh và mật khẩu mà sau đó có thể sử dụng để giả mạo vào hệ thống. Như được hiển thị bên dưới, ngay cả khi hệ thống xóa thông tin trước khi trả lại quyền kiểm soát cho chương trình của người dùng, người dùng có thể lấy thông tin thông qua một số hình thức xử lý đồng thời, chẳng hạn như thao tác I/O độc lập đọc từ không gian làm việc được đề cập. Có những biến thể khác của lỗ hổng này. Đôi khi các tệp công việc và không gian làm việc không bị xóa khi người dùng giải phóng chúng và người dùng khác có thể quét "bảng đen chưa được xóa" này khi không gian tệp hoặc không gian tạm thời chưa thể xóa được chỉ định tiếp theo.

Đôi khi các tác động đầy đủ của thông tin cung cấp cho người dùng không được các nhà thiết kế của hệ thống nhận ra. Ví dụ: các chương trình điều khiển thường xuyên xác nhận việc xử lý các yêu cầu dịch vụ của người dùng bằng cách đặt mã báo cáo lại/cờ trạng thái. Các điều kiện trả lại khác nhau (chẳng hạn như: "illegal parameter", "segment error", "password OK", v.v.) và các hình thức giao tiếp giữa các quá trình khác (ví dụ: xác nhận GỬI/NHẬN) có thể biểu thị thông báo cho phép người dùng bảo mật chặt chẽ.

## Xác thực bất đồng bộ

Tính toàn vẹn của hệ thống yêu cầu duy trì tính toàn vẹn của thông tin được truyền giữa các quá trình hợp tác hoặc trình tự chương trình điều khiển. Nếu tuần tự hóa không được thực thi trong cửa sổ thời gian giữa việc lưu trữ giá trị dữ liệu và tham chiếu của nó (hoặc giữa hai tham chiếu tuần tự), thì tính nhất quán của giá trị dữ liệu đó có thể bị phá hủy bởi một quy trình không đồng bộ.

Thông tin kiểm soát đặc biệt dễ bị sửa đổi bất cứ khi nào nó được đặt trong bộ nhớ mà quy trình cấp dưới có thể truy cập được. Đây đôi khi được gọi là vấn đề "kiểm tra thời gian sử dụng". Như được mô tả trong phần ngụ ý chia sẻ lỗ hổng dữ liệu đặc quyền, một hệ điều hành có thể thường xuyên chia sẻ không gian bộ nhớ với các chương trình người dùng. Không gian này có thể không chỉ được sử dụng để lưu trữ thông tin một cách thụ động mà còn có thể chứa các thông số hệ thống hoặc người dùng đại diện cho dữ liệu mà các hành động trong tương lai sẽ dựa trên đó. Bất cứ khi nào có "cửa sổ thời gian" giữa thời gian chương trình điều khiển xác minh một tham số và thời gian nó truy xuất tham số từ bộ nhớ dùng chung để sử dụng, một lỗ hổng bảo mật tiềm ẩn sẽ được tạo ra. Điều này là do các hệ điều hành hiện đại cho phép người dùng có hai hoặc nhiều hoạt động (quy trình) thực thi đồng thời và chia sẻ phân bổ bộ nhớ của người dùng đó. Ví dụ: người dùng có thể bắt đầu thao tác I/O và sau đó tiếp tục thực hiện chương trình của mình trong khi thao tác I/O hoàn tất.

Trong một ví dụ khác, người dùng chia sẻ thời gian có thể tạm thời tạm dừng một hoạt động bằng cách nhấn phím "chú ý" hoặc xác nhận tiêu cực (NAK) trên thiết bị đầu cuối của họ, thực hiện thao tác thứ hai, sau đó trả lại quyền điều khiển cho thao tác đầu tiên để hoàn thành. Một số hệ thống cho phép "đa nhiệm", trong đó hai hoặc nhiều chương trình đang chia sẻ bộ nhớ được chỉ định của một người dùng (không gian a-ddress) và đang thực thi đồng thời —có thể mỗi chương trình được thực thi đồng thời bởi CPU riêng biệt của một hệ thống máy tính đa xử lý.

Dưới đây là kịch bản một ví dụ về việc xác thực bất đồng bộ:

* Trong khung thời gian 1, người dùng đưa ra yêu cầu I/O cho chương trình điều khiển. Chương trình điều khiển xác nhận tất cả các tham số I/O (bao gồm cả con trỏ địa chỉ tới bộ đệm hợp lệ trong bộ nhớ được gán hợp pháp cho người dùng), xếp hàng yêu cầu I/O [phải đợi cho đến khi thiết bị thích hợp không còn bận), và sau đó trả lại quyền kiểm soát cho người dùng.
* Trong khung thời gian 2, người dùng thay thế con trỏ địa chỉ hợp lệ tới bộ đệm của mình bằng một địa chỉ trỏ đến một vị trí trong chương trình điều khiển.
* Khi I/O đang thực hiện trong khung thời gian 3, dữ liệu mà người dùng yêu cầu sẽ được đọc vào (hoặc ra khỏi) chương trình điều khiển thay vì bộ đệm hợp lệ của anh ta. Do đó, các hướng dẫn trong chương trình điều khiển có thể được phủ lên với các hướng dẫn do người dùng cung cấp hoặc thông tin về chương trình điều khiển đặc quyền có thể được đọc ra tệp của người dùng.

Trong một số hệ thống, chương trình điều khiển có thể sử dụng vùng lưu thanh ghi tràn, nằm trong vùng lưu trữ có thể truy cập của người dùng, bất cứ khi nào vùng lưu chính của chương trình điều khiển được lấp đầy. Thông tin đã lưu này thường chứa trạng thái chương trình và thông tin điều khiển.

Tình huống này có thể làm phát sinh một biến thể khác của lỗi xác thực không đồng bộ, nếu người dùng có thể sửa đổi thông tin điều khiển đó. Một ví dụ về nỗ lực thâm nhập như vậy như sau:

* Người dùng tạo một bản ghi I/O chỉ chứa một địa chỉ trỏ đến một vị trí mong muốn trong một trong các chương trình của người dùng.
* Sau đó, nhiều bản sao của bản ghi này sẽ được xuất ra dưới dạng tệp.
* Người dùng tiếp theo bắt đầu thao tác I/O để đọc các bản ghi này lặp đi lặp lại vào khu vực bộ nhớ của người dùng được chương trình điều khiển sử dụng làm bộ nhớ tràn cho các thanh ghi.
* Sau đó, người dùng đưa ra một yêu cầu dịch vụ hệ thống khiến chương trình điều khiển thực hiện một số lệnh gọi nội bộ màn hình lồng nhau, do đó làm tràn vùng lưu chính của nó. [Việc phát hành lặp lại các yêu cầu dịch vụ nhất định cũng có thể đạt được mục đích này.)
* Các thanh ghi được lưu bởi chương trình điều khiển trong vùng lưu tràn sẽ được phủ lên bởi các bản ghi đầu vào có chứa địa chỉ trỏ đến mã của người dùng. (Người dùng có thể cần một số điều chỉnh về thời gian để thực hiện điều này.)
* Khi chương trình điều khiển cuối cùng khôi phục các thanh ghi và trạng thái từ vùng tràn, nó sẽ chuyển quyền điều khiển sang chương trình của người dùng ở trạng thái giám sát/điều khiển - do đó cho phép người dùng toàn quyền kiểm soát hệ điều hành.

Hệ điều hành có thể lưu trữ thông tin trong một khoảng thời gian trong bộ nhớ phụ dùng chung cũng như trong bộ nhớ chính. Ví dụ: một hệ điều hành có thể có cung cấp điểm kiểm tra/khởi động lại để ghi lại trạng thái của một chương trình đang chạy tại các điểm khởi động lại thuận tiện dưới dạng kết xuất "điểm kiểm tra". Các kết xuất điểm kiểm tra này chứa cả dữ liệu người dùng và thông tin điều khiển chỉ định trạng thái điều khiển sẽ được chỉ định nếu chương trình được khởi động lại. Các kết xuất điểm kiểm tra được người dùng ghi lại trong một tệp được chỉ định cho hệ thống và người dùng đó có thể truy cập để thao tác. Thông qua thao tác như vậy, người dùng có thể khiến chương trình của mình được khởi động lại với thông tin trạng thái đã sửa đổi mang lại cho chương trình của anh ta các đặc quyền lớn hơn so với quy định ban đầu. Ví dụ, điều này có thể dẫn đến việc người dùng giành được quyền kiểm soát/giám sát trạng thái.

## Nhận dạng/ Xác thực/ Uỷ quyền không đầy đủ

Nhận dạng, ủy quyền và xác thực là các thành phần thiết yếu của khái niệm truy cập có kiểm soát. Sự ủy quyền — bản quyền truy cập được kiểm soát — đa phần dựa trên nhận dạng duy nhất, được xác thực của các cá nhân và tài nguyên. Một hệ điều hành về cơ bản là một trình quản lý tài nguyên. Do đó, một hệ điều hành phải đối mặt với các vấn đề về tính toàn vẹn bất cứ khi nào 1) nó không yêu cầu ủy quyền cho một cá nhân hoặc quy trình để truy cập vào bất kỳ dữ liệu nào hoặc sử dụng bất kỳ tài nguyên nào không phải có sẵn cho tất cả, hoặc 2) nó không xác định duy nhất các tài nguyên mà nó đang giải quyết.

Một lỗ hổng được tạo ra bất cứ khi nào hệ thống cho phép người dùng sở hữu một tập hợp các đặc quyền/khả năng bỏ qua hợp pháp các cơ chế bảo mật (truy cập có kiểm soát) và thực hiện một hành động chỉ được phép đối với những người dùng có các đặc quyền/khả năng khác nhau hoặc bất cứ khi nào nó cho phép tất cả người dùng thực hiện một hành động chỉ nên bị hạn chế đối với những người dùng có đặc quyền lớn hơn.

Một lỗ hổng nhận dạng/cách ly không đầy đủ có thể được tạo ra bất cứ khi nào một quy trình hệ thống dựa vào các cơ chế (được thực hiện ở nơi khác trong hệ thống) để đảm bảo sự cô lập của các tài nguyên hệ thống và do đó, tính đầy đủ của việc nhận dạng chúng. Đây có thể là một chính sách tồi nếu trên thực tế, các cơ chế không phù hợp.

Ví dụ, để được xác định duy nhất một chương trình phải được xác định bằng cả tên chương trình và tên của thư viện mà từ đó nó được tải. Nếu không, người dùng rất dễ tải trước một chương trình giả mạo có tên giống với một số quy trình chương trình điều khiển (phải được tải động khi cần thiết) và quy trình giả mạo này được chương trình điều khiển sử dụng thay cho quy trình xác thực .

Để thực hiện điều này, người dùng tạo một hoạt động dẫn đến chương trình điều khiển yêu cầu lại quy trình này. Trình tải sẽ thấy rằng quy trình được đặt tên (giả mạo) đã được tải (hợp pháp) và sẽ thiết lập chương trình kiểm soát để sử dụng chương trình giả mạo.

Ví dụ khác, cơ chế kiểm tra mật khẩu hoặc ID người dùng có thể bị phá vỡ nếu nó không giới hạn hiệu quả số lần người dùng có thể cố gắng đăng nhập vào hệ thống hoặc nếu nó không giới hạn thời gian đã trôi qua được phép hoàn tất đăng nhập. Trong những trường hợp như vậy, người dùng có thể sử dụng một máy tính khác để liệt kê đầy đủ tất cả các tổ hợp bit mật khẩu và do đó phá vỡ tính bảo mật của mật khẩu.

Một số hệ thống có tính năng kiểm tra ủy quyền mở rộng được liên kết với hầu hết, nhưng không phải tất cả, các phương thức truy cập tệp và không hạn chế việc sử dụng các phương thức truy cập không thực hiện kiểm tra ủy quyền. Bất kỳ người dùng nào có được tài liệu cho các phương pháp truy cập sau này (khả năng không cần thiết) chỉ cần sử dụng chúng để truy cập vào bất kỳ tệp nào trong hệ thống. Đây là một ví dụ về việc bỏ qua các cơ chế truy cập được kiểm soát.

Bảng 3-6 tóm tắt các danh mục và trình bày các ngoại lệ bổ sung của lỗ hổng nhận dạng, ủy quyền và xác thực không đầy đủ.

## Vi phạm giới hạn

Hệ điều hành được mô tả bằng cả cách thể hiện của nó trong hướng dẫn máy tính và bằng tài liệu bên ngoài của nó. Bất cứ khi nào hai mô tả này khác nhau, một lỗ hổng toàn vẹn có thể tồn tại. Một lỗ hổng bảo mật được tạo ra bất cứ khi nào giới hạn hệ điều hành được lập thành văn bản hoặc quy trình cấm theo quy trình không được thực thi.

Ví dụ: những người triển khai hệ điều hành không được xử lý tình huống đạt đến giới hạn trên về kích thước của bảng hoặc bộ đệm hoặc khi không gian hàng đợi trở nên bão hòa. Tài liệu có thể chỉ định chính xác các giới hạn trên và cấm vượt quá giới hạn, nhưng nếu người dùng cố tình hoặc vô tình gây ra tràn hoặc quá tải, thì các kết quả khác nhau có thể xảy ra - đôi khi có thể xảy ra sự cố hệ thống, đôi khi hoạt động của hệ thống bị suy giảm dữ liệu nhạy cảm có thể bị mất và trong một số trường hợp, dữ liệu đó có thể bị xâm phạm.

## Lỗi logic

Trong bất kỳ hệ điều hành chính nào, đều có - tại bất kỳ thời điểm nào - một số "lỗi" hoặc lỗi logic. Nhiều lỗi trong số này phụ thuộc vào các tình huống thời gian không thể thống kê được và không nằm trong tầm kiểm soát của bất kỳ người dùng cá nhân nào. Tuy nhiên, một số lỗi logic này có thể bị người dùng cố ý khai thác để làm tổn hại đến tính toàn vẹn của hệ thống.

Một ví dụ liên quan đến việc xử lý lỗi không chính xác. Ví dụ, hệ thống có thể thực hiện một hành động bất hợp pháp trước khi báo hiệu một điều kiện lỗi. Ví dụ, hãy xem xét rằng một người dùng yêu cầu một loạt các sửa đổi được thực hiện đối với mục nhập thư mục tệp của một người dùng khác, mà người dùng đầu tiên có quyền chỉ đọc. Nếu hệ thống thực hiện các hành động được yêu cầu và sau đó xác định rằng các hành động vượt quá sự cho phép của người dùng yêu cầu, thì tính bảo mật của hệ thống đã bị xâm phạm thông qua một lỗi logic. Điều này cũng có thể xảy ra nếu một dịch vụ hệ thống như kết xuất bộ nhớ được khởi tạo đồng thời với việc kiểm tra ủy quyền của người dùng để yêu cầu dịch vụ được chỉ định cho các khu vực lưu trữ được chỉ định. Vào thời điểm lỗi được phát hiện, các khu vực cấm có thể đã được liệt kê.

Ở mức độ tinh tế hơn, người dùng có thể phát hiện ra rằng các hướng dẫn số học nửa từ được sử dụng để xử lý không đúng tham số địa chỉ trả lại nửa từ. Nếu số nửa từ lớn nhất có thể được sử dụng làm địa chỉ, có thể xảy ra hiện tượng tràn không lường trước được, dẫn đến địa chỉ trỏ đến vị trí 0001 trong bộ nhớ chương trình điều khiển, điều này có thể gây ra sự cố hệ thống.

Trong một tình huống khác, bằng cách nhấn vào nút ngắt "chú ý" (hoặc NAK) trên thiết bị đầu cuối của anh ấy trong quá trình in thông báo lỗi đăng nhập, người dùng có thể khiến hệ thống chấp nhận sai một nỗ lực đăng nhập mới mà không tiến bộ đếm được đặt thành ghi lại số lần đăng nhập trước đó. Lỗi hệ thống này cho phép tự động hóa việc liệt kê đầy đủ các mật khẩu, không có dấu hiệu cho người vận hành hệ thống biết rằng điều này đang diễn ra.

Trong một ví dụ cuối cùng về xử lý lỗi không chính xác, đôi khi xảy ra trường hợp cơ chế bảo vệ bị vô hiệu hóa hoặc sửa đổi do lỗi người dùng (cố ý) và không thể đặt lại khi chương trình điều khiển sau đó trả lại quyền điều khiển cho người dùng. Điều này có thể dẫn đến việc người dùng nhận được các đặc quyền trái phép.

Có thêm hai loại lỗi logic có thể khai thác. Những điều này được liệt kê ở đây cho đầy đủ và không có ví dụ:

* Bắt đầu hoặc kết thúc quá trình/chức năng không chính xác.
* Bẫy lỗi phần mềm trạng thái kiểm soát.

# KẾT LUẬN

## Kết luận chung

Xem Mục 1.3.12

## Hướng phát triển

(Nếu có)

## Kiến nghị và đề xuất

(Nếu có)

# TÀI LIỆU THAM KHẢO

1. T. H. Cormen, C. E. Leiserson, and R. L. Rivet, *Introduction to Algorithm*. MIT Press, McGraw-Hill, 1990.
2. J. W. DuBois, S. Schuetze-Coburn, S. Cumming, and D. Paolino, “Outline of discourse transcription,” in *Talking Data: Transcription and Coding in Discourse Research*, J. A. Edwards and M. D. Lampert, Ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, pp. 45-89.
3. J. M. Airey, J. H. Rohfl, F. Brooks Jr., “Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments,” *Comptuer Graphics*, Vol. 24, No. 2, pp. 41-50, 1990.
4. S. Brandt, G. Nutt, T. Berk, M. Humphrey, “Soft Real time Application Execution with Dynamic Quality of Service Assurance,” in *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service*, Hawaii, USA, May 1998, pp. 154-163.
5. K. Riley, “Language theory: Applications versus practice,” presented at the Conf. of the Modern Language Association, Boston, MA, December 27-30, 1990.
6. J. Jones. (1991). *Networks* (2nd ed.) [Online]. Available: http://www.atm.com.