TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

Đề tài:

SELINUX VÀ ỨNG DỤNG VÀO BẢO MẬT DỮ LIỆU TRONG DOANH NGHIỆP

Sinh viên thực hiện:

CÙ XUÂN TOẢN **ĐTVT12 – K54**

NGUYỄN THANH NHÀN **ĐTVT04 – K54**

Giảng viên hướng dẫn: **ThS. VŨ SONG TÙNG**

Hà Nội, 06.2014

LỜI MỞ ĐẦU

Ngày nay, sự phát triển mạnh mẽ của công nghệ đã tạo những điều kiện thuận lợi cho các lĩnh vực kinh doanh, sản xuất và đời sống xã hội. Và đặc biệt trong nền kinh tế công nghiệp hóa, hiện đại hóa thì mối quan hệ tương hỗ này càng trở nên quan trọng và ý nghĩa. Bên cạnh những mặt thuận lợi các công nghệ mang lại, có rất nhiều mối đe dọa tiềm ẩn và điển hình trong số đó là nguy cơ về mất an toàn, bảo mật thông tin dữ liệu. Do đó, việc xây dựng hàng rào kỹ thuật để ngăn chặn những truy cập trái phép trở thành nhu cầu cấp bách trong các hoạt động bảo mật thông tin.

Một hệ thống bảo mật dữ liệu được đánh giá cao - SELinux - đã được triển khai trong nhiều bản Linux, mang lại những bước tiến mới trong lĩnh vực bảo mật nói chung và điều khiển truy cập nói riêng. Với đề tài ***"SELinux và ứng dụng vào bảo mật dữ liệu trong doanh nghiệp"*** đồ án này sẽ đi sâu nghiên cứu cấu trúc bên trong cũng như nguyên lý hoạt động của SELinux, đưa ra các ví dụ nhằm chứng minh vai trò cũng như tính hiệu quả của phương thức bảo mật này. Từ đó áp dụng SELinux để xây dựng hệ thống bảo mật thông tin, dữ liệu trong môi trường thực tế với nhu cầu bảo mật thông tin cao, đó là những doanh nghiệp.

Chúng tôi xin trân trọng bày tỏ lòng biết ơn sâu sắc đên đến Thầy giáo, **ThS. Vũ Song Tùng** - Bộ môn Điện tử và kỹ thuật máy tính - Viện Điện Tử Viễn Thông – Đại Học Bách Khoa Hà Nội, các thầy cô trong bộ môn, cùng với anh **Vũ Thành Công**, anh **Nguyễn Đình Thắng**, anh **Nguyễn Đăng Khoa** - Trung tâm nghiên cứu và phát triển Mobile Samsung Việt Nam (SVMC) – đã tận tình hướng dẫn, giúp đỡ chúng tôi hoàn thành đồ án này.

Chúng tôi xin chân thành cảm ơn!

Hà Nội, tháng 6 năm 2014

TÓM TẮT ĐỒ ÁN

Trong một nền kinh tế toàn cầu hóa như hiện nay thì vấn đề an toàn thông tin được xem là sự sống còn đối với các doanh nghiệp. Thế nhưng, rất nhiều doanh nghiệp vẫn chưa nhận thức được tầm quan trọng của vấn đề bảo mật thông tin và những nguy cơ có thể xảy ra từ việc rò rỉ thông tin trong chính nội bộ của doanh nghiệp mình. Chính vì thế, việc áp dụng các hệ thống quản lý bảo mật thông tin là việc làm rất quan trọng của các doanh nghiệp trong thời kỳ nền kinh tế hội nhập toàn cầu hóa như hiện nay.

Trong đề tài này, SELinux sẽ được đi sâu tìm hiểu về cấu trúc cùng với nguyên lý hoạt động và từ đó giải pháp cho bảo mật thông tin trong các doanh nghiệp bằng việc sử dụng hệ thống SELinux sẽ được giới thiệu. Đồng thờiđể thực hiện giải pháp này, cơ cấu tổ chức, mô hình hoạt động của các doanh nghiệp nói chung và loại hình công ty cổ phần nói riêng sẽ được khảo sát và phân tích. Để làm nổi bật được hoạt động của SELinux và sự khả thi của giải pháp này, những lý thuyết chung về SELinux sẽ được trình bày rồi sau đó việc kiểm soát truy nhập thông tin, dữ liệu của doanh nghiệp thông qua hệ thống máy tính Client – Server được tiến hành. Server sẽ là máy lưu giữ những dữ liệu cần được bảo mật. Còn Client là máy tính của những thành viên trong công ty muốn truy cập vào máy Server để thực hiện các công việc nào đó với những dữ liệu được lưu trữ như đọc, sửa, hoặc xóa các thông tin. Để thực hiện bảo mật bằng SELinux, các policy sẽ được xây dựng và tích hợp vào toàn bộ hệ thống máy tính trong doanh nghiệp.

Tuy nhiên do những hạn chế trong thời gian và kinh nghiệm nghiên cứu nên đồ án mới dừng lại ở những kết quảmang tính chất chứng minh, chưa thể áp dụng được vào hệ thống lớn và quy mô thực hiện cũng không hoàn thiện như trong một doanh nghiệp thực. Những hạn chế và những ý tưởng, mục tiêu còn chưa hoàn thiện sẽ là những khắc phục và hướng phát triển tiếp theo trong tương lai của đề tài.

ABSTRACT

In today's globalized economy, information security is considered one of the most important things to enterprises. However, a large number of companies have not realized the importance of information security as well as risks that may happen due to leak of information from internal employees of these companies'. Therefore, that applying information security management system is very crucial, especially in today's globalized economy.

In this thesis, a solution for business information security by using SELinux security system is going to be presented. Knowing SELinux as the most effective security solution enhanced for recent Linux versions, SELinux was being researched to more understand about its' architecture and operation. Additionally, the organization structure, operation model of a type of companies – Join stock company – the typical company for business information security demand was also being analyzed and summarized. To highlight SELinux operation and its practicability, firstly the fundamental theories about SELinux will be presented, then that is deep description of the waywhich SELinux controls accessibility using Clients – Server model. The Server is a computer which saves all information, data needed to protect and digitalized policies. The Clients are computers of company members which want to make access to the Server for their work such as read, update or delete information, data.

However, due to limited time, the thesis project is just finished with simple simulations which are unable to apply to real system and big companies. However, it is certainly that the restriction will be proved and the idea of thesis's security solution will be continued developing in the future.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH SÁCH CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Tên đầy đủ
1	SVMC	Samsung Vietnam Mobile R&D Center
2	WIPO	World Intellectual Property Organization
3	SHTT VN	Sở hữu trí tuệ Việt Nam
4	IT	Information Technology
5	CEO	Chief operations officer
6	MLS	Multi-levels Security
7	SELinux	Security Enhanced Linux
8	IP	Internet Protocol
9	NSA	National Security Agency
10	DNS	Domain Name System
11	DAC	Discretionary Access Control
12	MAC	Mandatory Access Control
13	RBAC	Role-bases Access Control
14	TE	Type Enforcement
15	MLS	Multi-levels Security
16	MCS	Multi-Categories Security
17	ACL	Access Control Lists
18	ACM	Access Control Matrix
19	AVC	Access Vector Cache
20	HĐQT	Hội đồng quản trị
21	TGĐ	Tổng giám đốc

Chương 1. Tổng quan về bảo mật tài liệu trong doanh nghiệp

Ngày nay, vấn đề bảo hộ kinh doanh đang được các doanh nghiệp hết sức quan tâm do nhiều yếu tố. Trong đó, mức độ cạnh tranh trên thị trường ngày càng khốc liệt nên các đối thủ chắc chắn không hề muốn chia sẻ thông tin cho nhau. Hơn nữa, khi mà người lao động có quyền tự do lựa chọn và thay đổi nơi làm việc, sẽ có khả năng rất cao họ mang theo thông tin đến nơi làm việc mới mà thông thường lại là đối thủ cạnh tranh của công ty cũ. Cuối cùng, do bản thân các thông tin bí mật không phải là giải pháp kỹ thuật nên không thể được bảo hộ dưới danh nghĩa sáng chế; và cũng không thể giải trình công khai để đăng ký bảo hộ do tính bảo mật của thông tin. Do vậy, nhu cầu bảo mật thông tin trong các doanh nghiệp là rất cao, đòi hỏi phải có hệ thống bảo mật toàn diện, chặt chẽ. Tuy nhiên nhiều doanh nghiệp vẫn chưa thực sự quan tâm, kiểm soát và đầu tư cho vấn đề trọng yếu này, dẫn tới nhiều hậu quả nghiêm trọng như rò rỉ thông tin do vấn đề xử lý tài liệu thiếu cẩn trọng, hoặc bị chính những nhân viên hoặc những người đã từng là nhân viên của công ty mình tiết lộ ra ngoài. Đây là những nguy cơ vô cùng tai hại, vì vậy mà bảo mật thông tin, dữ liệu cần được quan tâm và đầu tư hàng đầu như một yếu tố sống còn của doanh nghiệp.

1.1 Những vấn đề trong bảo mật ở doanh nghiệp

Mỗi công ty, doanh nghiệp đều có những thông tin, dữ liệu mật cần phải bảo vệ. Vì nếu những thông tin này rò rỉ ra ngoài, đặc biệt là rơi vào tay của các đối thủ thì hậu quả sẽ hết sức nghiêm trọng. Những thông tin, dữ liệu này được nhắc tới qua thuật ngữ "bí mật kinh doanh". Theo định nghĩa của Cục sở hữu trí tuệ Việt nam, "bí mật kinh doanh là thông tin thu được từ hoạt động đầu tư tài chính, trí tuệ, chưa được bộc lộ và có khả năng sử dụng trong kinh doanh [1]."

Cũng theo Cục SHTT VN [1], bí mật kinh doanh được bảo hộ nếu đáp ứng các điều kiện sau đây:

a) Không phải là hiểu biết thông thường và không dễ dàng có được;

b) Khi được sử dụng trong kinh doanh sẽ tạo cho người nắm giữ bí mật kinh doanh lợi thế so với người không nắm giữ hoặc không sử dụng bí mật kinh doanh đó;

c) Được chủ sở hữu bảo mật bằng các biện pháp cần thiết để bí mật kinh doanh đó không bị bộc lộ và không dễ dàng tiếp cận được.

Bí mật kinh doanh có thể liên quan đến các loại thông tin khác nhau như kỹ thuật và khoa học (công thức sản xuất, cấu tạo kỹ thuật, mã máy tính, dữ liệu thử nghiệm…); thương mại (danh sách các nhà cung cấp và khách hàng, các chiến lược tiếp thị, quảng cáo và kinh doanh, kết quả nghiên cứu thị trường, phương pháp bán hàng…); tài chính (cơ cấu giá nội bộ, danh mục giá…); thông tin phủ định (tình trạng bế tắc trong nghiên cứu, các giải pháp kỹ thuật đã bị rút bỏ)….

Tuy nhiên có một vấn đề nan giải cho các doanh nghiệp là bí mật kinh doanh không thể được bảo hộ để chống lại việc tìm ra các thông tin này theo các cách công bằng và trung thực, như sử dụng phương pháp phân tích ngược hoặc sử dụng một biện pháp độc lập nào đó. Khi một người không có quyền tiếp cận một cách hợp pháp những thông tin bí mật kinh doanh nhưng lại giải mã được các thông tin đó mà không sử dụng bất kì phương tiện bất hợp pháp nào thì người đó không thể bị ngăn cấm sử dụng thông tin đã được tìm ra. Trong những trường hợp như vậy, chủ sở hữu bí mật kinh doanh không thể thực hiện bất kì một hành động pháp lý nào chống lại người này.

Giải pháp có thể được các doanh nghiệp tìm đến là đăng kí độc quyền sáng chế để các sáng chế bí mật của họ được bảo hộ bởi pháp luật. Tuy nhiên, không phải bất kì thông tin nào cũng được bảo hộ, chưa kể chi phí và thời gian đăng kí cũng là bài toán đặt ra. Không chỉ vậy, bảo hộ sáng chế còn có những hạn chế như: gây tổn thất cho xã hội và không khuyến khích sự chia sẻ tri thức. Đồng thời, việc bảo hộ sáng chế ngăn cản sự kết hợp các ý tưởng để tạo ra một phát minh, sáng chế mới [2]. Thêm vào đó, thời gian tối đa cho bảo hộ sáng chế là 20 năm, vậy thì, sau 20 năm đó thì sáng chế đó sẽ trở thành tài sản chung của nhân loại [3]. Như vậy thì vấn đề bảo mật sáng chế cũng

chưa thể giải quyết triệt để chứ chưa nói tới bảo mật thông tin, tài liệu bí mật kinh doanh của doanh nghiệp.

Một ví dụ thú vị về việc bảo mật thông tin, đó là công thức chế biến đố uống nhẹ mang tên Coca Cola - một bí mật kinh doanh của công ty Coca Cola. Chỉ một vài người trong công ty biết được công thức này; và nó được giữ bí mật trong một chiếc hầm của một ngân hàng ở Atlanta, bang Georgia; những người biết được công thức bí mật này đã ký hợp đồng không tiết lộ [4]. Chính vì quyết định giữ bí mật về công thức này thay vì đăng ký cấp bằng sáng chế, đến nay, công ty Coca Cola vẫn là doanh nghiệp duy nhất có thể sản xuất được loại nước uống đặc biệt được toàn cầu ưa chuộng. Còn nếu công thức này được cấp bằng sáng chế (chỉ đươc bảo hộ tối đa là 20 năm, sau đó sẽ trở thành tài sản chung của nhân loại), sau 20 năm mọi thành phần và công đoạn chế biến Coca Cola sẽ được bộc lộ công khai, và cả thế giới đều có thể sản xuất Coca Cola.

Quay lại vấn đề bảo mật thông tin trong các doanh nghiệp, theo tổ chức trí tuệ thế giới WIPO, có tới 10 biện pháp được đề xuất để hạn chế tối thiểu thiệt hại do rò rỉ bí mật kinh doanh [5]. Theo đó thì đây không chỉ là vấn đề, nhiệm vụ của phòng thông tin hay hệ thống IT trong doanh nghiệp, mà còn liên quan đến cơ cấu, tổ chức, trách nhiệm của cả bộ máy công ty. Tuy nhiên, do phạm trù và quy mô nghiên cứu, một số biện pháp kĩ thuật hiện nay đang được sử dụng để bảo mật thông tin trong các công ty, tập đoàn lớn sẽ được giới thiệu trong đề tài này. Và cũng vì phạm vi nghiên cứu và ứng dụng là nội bộ doanh nghiệp trong cùng một không gian làm việc, do đó đề tài sẽ bỏ qua khía cạnh bảo mật trên đường truyền mà chú trọng vào giải pháp bảo mật ngay trong hệ thống dữ liệu của doanh nghiệp.

Hiện nay giải pháp mà các doanh nghiệp sử dụng để bảo mật dữ liệu thường là các biện pháp mã hóa dữ liệu bằng phần mềm chuyên dụng. Phương pháp này có hạn chế là tất cả các dữ liệu của doanh nghiệp sẽ bị mã hóa, dẫn tới phức tạp, rắc rối trong

việc truyền gửi những dữ liệu không phải là bí mật ra ngoài. Hoặc giải pháp bảo mật Office 365 của Microsoft trên Windows sử dụng cơ chế điều khiển truy nhập dựa trên cơ sở vai trò (RBAC – Role-base Access Control). Tuy nhiên giải pháp này lại dẫn tới sự thiếu linh hoạt vì bị phụ thuộc vào hệ điều hành mà vẫn chưa đảm bảo được sự bảo mật cao do chỉ kiểm soát dựa trên vai trò, tức là công việc, vị trí, trách nhiệm của thành viên trong doanh nghiệp.

Một giải pháp nữa là phân quyền dữ liệu để kiểm soát truy nhập của các thành viên trong doanh nghiệp, giải pháp này sẽ được trình bày chi tiết ở những phần tiếp theo của báo cáo này. Mặt khác, nhiều doanh nghiệp, thậm chí các tổ chức chính phủ đã sử dụng phương cách bảo mật thông tin bằng việc bo vùng dữ liệu sử dụng cho các ứng dụng độc quyền, ví dụ như giải pháp Knox của Samsung. Giải pháp Knox tưởng chừng như hoàn hảo, tuy nhiên gần đây nhiều lỗ hổng bảo mật cũng đã được tìm ra ở giải pháp này [6].

1.2 Mô hình kiểm soát truy cập ở các doanh nghiệp.

Trong một doanh nghiệp, đặc biệt đối với những doanh nghiệp sản xuất hoặc dịch vụ thì với một lượng dữ liệu lớn về sản xuất và thông tin khách hàng cần được bảo mật thì nhu cầu về an ninh thông tin, dữ liệu càng mang tính chất quan trọng, sống còn đối với doanh nghiệp đó. Ví dụ như một ngân hàng, chỉ riêng việc ngân hàng phải lưu một khối lượng dữ liệu khổng lồ về khách hàng đã đủ nói nên nhu cầu phải làm sao bảo mật được những thông tin này. Bởi vì những thông tin này không chỉ quan trọng đối với ngân hàng mà còn quan trọng đối với chính khách hàng, ngân hàng sẽ phải chịu trách nhiệm trước pháp luật nếu làm lộ những thông tin này.

Thông thường, một doanh nghiệp cần đảm bảo bảo mật cho những loại thông tin, dữ liệu sau:

- Các thông tin, tài liệu nội bộ như kế hoạch đầu tư, phương hướng phát triển, chiến lược kinh doanh, sản xuất sản phẩm, thị trường mục tiêu, khách hàng đối tác quan trọng [7]…
- Các thông tin khách hàng đang giao dịch, nhà cung cấp…
- Các bí quyết kinh doanh như công thức chế tạo sản phẩm, quy trình công nghê chế tạo, thiết kế…
- Kế hoạch và phương pháp đào tạo nhân viên, phát triển chuyên môn…
- Thông tin tài chính: cơ cấu giá nội bộ, danh mục giá…
- Thông tin phủ định: tình trạng bế tắc trong nghiên cứu, phát triển [8]…

Như đã nói ở phần trước, bảo đảm an toàn thông tin và bảo mật bí mật kinh doanh không chỉ là nhiệm vụ và trách nhiệm của riêng phòng IT hay một nhóm nào đó vận hành hệ thống bảo mật cho doanh nghiệp mà nó cần có sự hợp tác, phối hợp của cả doanh nghiệp. Sẽ rất tai hại nếu như cơ chế quản lý truy xuất thông tin quá lỏng lẻo, thậm chí là dễ dãi. Và cũng có những thông tin, dữ liệu, tài liệu mà không phải mọi thành viên trong doanh nghiệp đều được tiếp cận, sửa, xóa. Do đó phân quyền truy cập thông tin là giải pháp đầu tiên, cơ bản nhưng lại rất hiệu quả được áp dụng ở các doanh nghiệp hiện nay.

Sẽ chẳng có vấn đề gì cần phải quan tâm nếu như doanh nghiệp đưa ra một quảng cáo mới cho sản phẩm, hoặc một câu khẩu hiệu sản xuất mới. Và tất cả mọi người từ các thành viên CEO cho tới các nhân viên, thậm chí là người ngoài công ty cũng đều được biết. Nhưng nếu là một bản phương hướng phát triển sản xuất mới, hoặc là bản danh sách đối tác chiến lược… thì vấn đề phân quyền truy cập vào những dữ liệu này sẽ rất được quan tâm. Vì đó là những thông tin nhạy cảm, ảnh hưởng tới sự phát triển sống còn của doanh nghiệp.

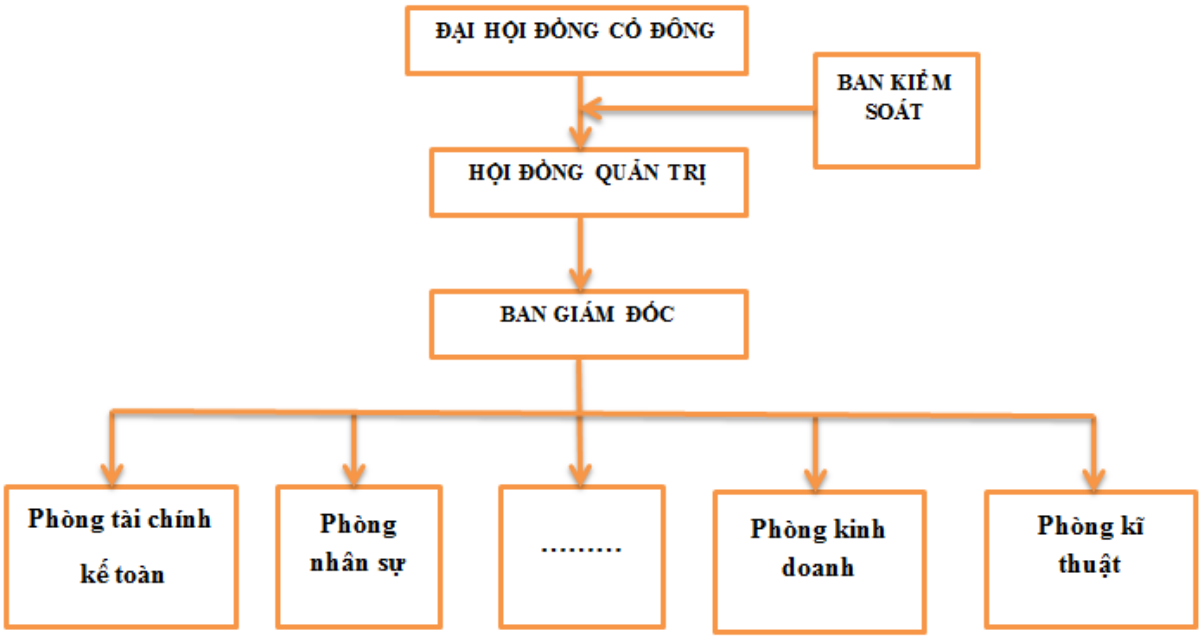
Phân quyền ở đây cần xét tới 2 khía cạnh, một là từ phía người chủ động truy cập (Subject), hai là từ phía dữ liệu bị truy cập (Object). Có rất nhiều loại hình doanh

nghiệp và cơ cấu tổ chức của một doanh nghiệp cũng rất phức tạp. Nhưng để minh họa cho giải pháp phân quyền đang được đề cập tới, một mô hình cơ cấu rút gọn của loại hình công ty cổ phần sẽ được sử dụng như đại diện cho các doanh nghiệp cần được bảo mật. Mô hình cơ cấu tổ chức rút gọn của loại hình công ty Cổ phần được trình bày trong hình 1.1.

Theo giải pháp phân quyền thông thường thì có hai cách là phân quyền trực tiếp cho từng người dùng (User) và phân quyền cho một nhóm người dùng (Group). Toàn bộ hệ thống được quản lý bởi Administrator (Admin - người quản trị), tùy theo chức năng của hệ thống và chức năng công việc của từng bộ phận, phòng ban, từng thành viên mà Admin sẽ phân quyền cho từng nhóm người hoặc từng người khác nhau. Việc phân quyền cho nhóm tạo ra sự linh hoạt trong nhóm đó, mọi người trong nhóm đều có quyền truy cập vào một loại dữ liệu nào đó mà nhóm khác nếu không được phân quyền tương tự thì sẽ không được phép truy cập vào. Ví dụ như phòng kế toán, tùy vào công việc mà một hoặc một nhóm các thành viên trong phòng kế toán sẽ được phép xem các tài liệu liên quan tới thu, chi, công nợ, thuế, lương…Khi đó nhóm người cùng làm một công việc, giải sử là tính toán thu, chi, thì sẽ được coi là một nhóm và nhóm này được cấp quyền xem (Read), cập nhật (Write), thậm chí là xóa (Delete) đối với những dữ liệu liên quan. Riêng Kế toán trưởng sẽ là người phê duyệt những tài liệu cần thiết. Rõ ràng khi đó Kế toán trưởng chỉ có quyền xem những dữ liệu này rồi phê duyệt hoặc thông báo cho nhân viên/nhóm nhân viên phụ trách nếu có sai sót gì, ngoài ra không có quyền cập nhật hay xóa những dữ liệu đó. Khi đó thì sự phân quyền sẽ phải chi tiết đến mức người dùng.

Khái niệm “phân quyền” được nói tới khi xét về quyền (Permission) mà một chủ thể (Subject) được tác động đến một đối tượng (Object). Ví dụ ông A được đọc/sửa file X. Nhưng khi xét về loại đối tượng thì cần một cơ chế nữa là phân cấp. Hai User có cùng quyền nhưng có thể có cấp khác nhau. Khái niệm “phân cấp” tạo ra sự chặt chẽ

hơn cho cơ chế bảo mật. Ví dụ ông A và ông B cùng có quyền đọc đối với file X. Nhưng do cấp của ông A và B khác nhau nêu chỉ ông B mới được quyền ghi đối với file X này. Nhắc tới khái niệm "Phân cấp" dẫn tới sự tương đồng với cơ chế an ninh đa cấp độ (MLS – Multi-levels Security).



Hình 1. 1. Mô hình cơ cấu tổ chức đơn giản của loại hình công ty cổ phần

Khi nhắc tới phân cấp thì không chỉ cần phân cấp cho chủ thể (Subject) mà còn cần phân cấp cho đối tượng bị truy cập (Object). Mối quan hệ giữa cấp độ của chủ thể với đối tượng sẽ quyết định quyền truy cập có được thực hiện không. Tuy nhiên nguyên lý ra sao thì còn tùy vào từng hệ thống. Cơ chế MLS sẽ là một minh họa về nguyên lý này và chi tiết về cơ chế này sẽ được trình bày ở những phần tiếp theo trong báo cáo này.

Giải pháp phân quyền và phân cấp là giải pháp được sử dụng phổ biến hiện nay trong các hệ thống bảo mật dữ liệu. Tuy nhiên điều đáng nói ở đây là trong hệ điều hành Linux từ sau phiên bản 2.4.x đã tích hợp sẵn một công cụ hay nói cách khác là một hệ thống bảo mật hiệu quả là SELinux (Security Enhanced Linux) [10]. SELinux

sẽ đáp ứng được những yêu cầu về phân quyền, phân cấp và còn hơn thế nữa cho một cơ chế bảo mật dữ liệu hoàn hảo và sẽ được đề cập tới ở phần kế tiếp.

1.3 Nguy cơ truy cập trái phép và rò rỉ thông tin

Lưu trữ và quản lý tốt kho dữ liệu luôn là yếu tố tiềm ẩn quyết định sự thành công và sống còn của doanh nghiệp. Lưu trữ dữ liệu tưởng chừng là một vấn đề đơn giản đối với mỗi cá nhân sử dụng máy tính cho nhu cầu học tập, giải trí của mình, nhưng đối với một doanh nghiệp có quy mô lớn thì lại là một vấn đề phải cân nhắc. Yêu cầu đặt ra cho việc quản lý lưu trữ gồm hai yếu tố là dung lượng lưu trữ, tốc độ truy cập và tính an toàn bảo mật thông tin, dữ liệu.

Để đáp ứng nhu cầu lưu trữ dữ liệu của các doanh nghiệp, rất nhiều phương thức lưu trữ đã được đưa ra, với những ưu và nhược điểm khác nhau, do đó tính khả dụng cũng tùy thuộc vào từng nhu cầu và quy mô của công ty, doanh nghiệp. Những giải pháp lưu trữ được sử dụng phổ biến hiện nay:

- **Sử dụng các ổ đĩa lưu động hoặc các ổ cứng ngoài**: giải pháp này phù hợp với những công ty có quy mô nhỏ, thậm chí là những cá nhân thuộc công ty có nhu cầu sao lưu, sử dụng dữ liệu khi đi công tác xa. Rõ ràng giải pháp này không giải quyết được triệt để vấn đề về dung lượng và tính bảo mật dữ liệu.
- **Lưu trữ trên mạng trực tuyến**: Các doanh nghiệp cũng có thể khai thác các lợi ích của các dịch vụ cung cấp dung lượng lưu trữ và sao lưu dữ liệu trên hệ thống Internet. Và trong phần lớn các trường hợp, bất kì một ai biết mật khẩu cũng có thể sử dụng trình duyệt web để truy cập và lấy các tệp dữ liệu ngay cả khi không ngồi tại công ty.. Giải pháp này khá đơn giản nhưng có nhiều lỗ hổng bảo mật và tốc độ truy cập bị phụ thuộc vào tốc độ đường truyền.
- **Lưu trữ trên mạng máy tính với mô hình Client-Server**: Lưu trữ trên mạng máy tính là cách lưu giữ số liệu đơn giản, dễ dàng truy cập với tốc độ

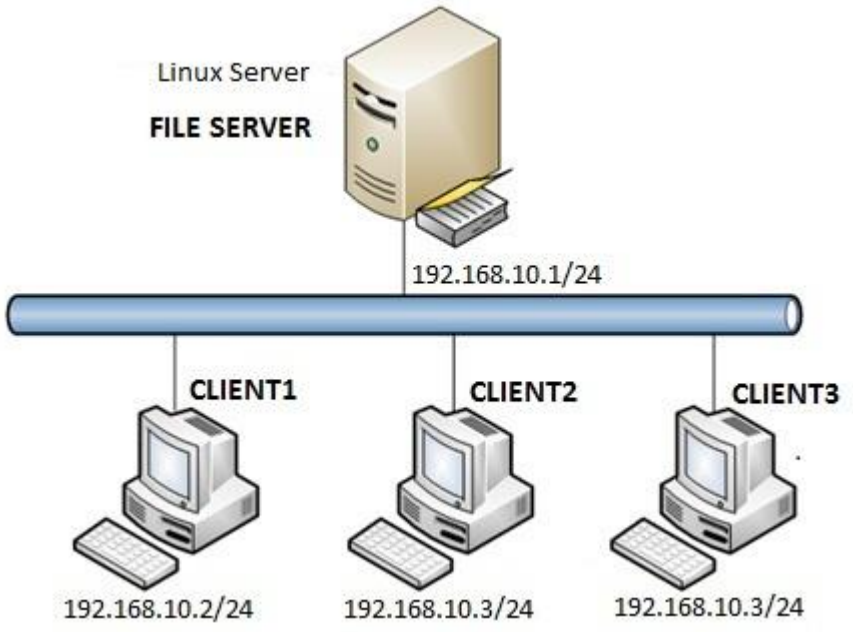
nhanh và đáng tin cậy. Giải pháp này phù hợp với các doanh nghiệp luôn có nhu cầu phải sử dụng các tệp dữ liệu về thông tin kinh tế có dung lượng lớn, nhiều người sử dụng cùng lúc và cho phép chia sẻ được trong hệ thống mạng máy tính nội bộ trong doanh nghiệp. Trong một chừng mực nào đó thì giải pháp này đáp ứng được tính bảo mật và dung lượng lưu trữ cũng như tốc độ truy cập.

Với những lợi thế so với những phương pháp lưu trữ và bảo mật khác thì mô hình mạng Client-Server hiện nay vẫn được sử dụng phổ biến trong các doanh nghiệp. Theo đó, hệ thống máy tính trong doanh nghiệp sẽ gồm một máy Server và các máy Client. Máy Server được xem như một "kho dữ liệu chung" của toàn doanh nghiệp, còn các máy Client chính là máy làm việc của mỗi thành viên trong công ty. Tất cả các máy Client sẽ được kết nối tới máy Server và được cấp quyền truy cập trong một giới hạn nào đó. Hình 1.2 mô tả mô hình quản lý và chia sẻ dữ liệu Client-Server trong các doanh nghiệp hiện nay.

Theo như phương pháp lưu trữ Client-Server trên mạng máy tính của doanh nghiệp thì dữ liệu sẽ được lưu trữ ở hai nơi là máy Server và các máy Client. Máy Server lưu trữ các dữ liệu chung, phục vụ cho nhiều người, trong khi các máy Client chứa các file dữ liệu riêng ứng với công việc, vai trò của thành viên đó trong doanh nghiệp. Giải pháp lưu trữ này tưởng chừng đáp ứng tốt tính bảo mật dữ liệu nhưng điều đó chưa hoàn toàn đúng bởi những lí do sau:

Thứ nhất, trong thời kì các trang web xã hội được ưa chuộng như ngày nay, chỉ cần với vài thao tác đơn giản là dữ liệu có thể được đưa ra ngoài, thậm chí được hiển thị công khai cho tất cả mọi người cùng xem. Chưa kể tới có quá nhiều cách để có thể gửi hoặc mang được dữ liệu ra ngoài như sử dụng các hòm thư điện tử như Google mail, Yahoo mail…, các ứng dụng giao tiếp thoại, phi thoại như chat Skype, Yahoo,

Hangout… Thậm chí là việc sử dụng các thiết bị lưu trữ ngoài như USB, thẻ nhớ, hoặc sử dụng điện thoại để ghi lại những thông tin nhạy cảm trong doanh nghiệp là điều khó tránh khỏi. Trước những nguy cơ này, nhiều doanh nghiệp đã áp dụng những biện pháp nghiêm ngặt như kiểm soát nhân viên ra/vào qua cửa từ, hoặc các công nghệ phát hiện thiết bị lưu trữ điện tử; chặn các trang web xã hội như Gmail, Facebook…; dán tem che camera điện thoại khi vào công ty…



Hình 1. 2. Mô hình lưu trữ dữ liệu Client-Server trong doanh nghiệp.

Thứ hai, mặc dù doanh nghiệp thực hiện nhiều biện pháp nghiêm ngặt để quản lý nhân viên của mình, nhưng nhiều khi chính nhân viên lại không hề biết mình đã vô tình gây rò rỉ thông tin khi cài đặt một số phần mềm mà tin tặc hoặc các hacker đã tích hợp kèm theo các viruts hoặc các chức năng nhằm ăn cắp thông tin. Nguy cơ này rất tai hại, một số doanh nghiệp chủ động ngăn chặn bằng việc cài đặt máy tính trước khi giao cho nhân viên sử dụng, và cấm họ cài đặt bất kì một phần mềm nào trong quá trình sử dụng mà chưa được phép. Tuy nhiên, vẫn có những sai phạm vô tình hay cố ý từ phía nhân viên, cách mà các doanh nghiệp áp dụng là trừng phạt nhân viên bằng một hình thức nào đó, và trong hầu hết các trường hợp thì đầu tiên nhân viên sẽ bị khóa địa chỉ IP.

Thứ ba, nếu không có biện pháp bảo mật dữ liệu trên Server thì một gián điệp được cài vào doanh nghiệp sẽ có thể xem những tài liệu quan trọng, và điều này gây ra hậu quả vô cùng nghiêm trọng cho doanh nghiệp.

Như vậy, mặc dù có rất nhiều chính sách, biện pháp để hạn chế việc rò rỉ thông tin trong doanh nghiệp. Nhưng mô hình lưu trữ Client-Server – được sử dụng phổ biến nhất hiện nay – vẫn tiềm ẩn những rủi ro lớn mà nếu không được khắc phục thì sẽ gây tổn hại lớn cho doanh nghiệp. Tuy nhiên ngay từ khi bắt đầu, thuật ngứ "SELinux" đã được nhắc tới gắn với giải pháp bảo mật dữ liệu cho doanh nghiệp. Giải pháp này là một đề xuất mới từ đề tài, mang tính mới mẻ nhưng lại khả thi và tính khả thi của nó sẽ dần được chứng minh trong những phần tiếp theo của đồ án.

1.4 Tổng quan về SELinux

- ***Khái niệm***

SeLinux (Security Enhanced Linux) là cơ chế điều khiển truy cập được tích hợp vào các phiên bản Linux từ 2.4. SELinux xây dựng lên một cơ chế ràng buộc (security enforcement) để kiểm soát các quá trình truy cập chặt chẽ hơn thông qua hệ thống chính sách bảo mật (security policy) được định nghĩa dựa trên các khái niệm của các mô hình Access control như MAC, RBAC, TE...

- ***Lịch sử phát triển của Selinux***

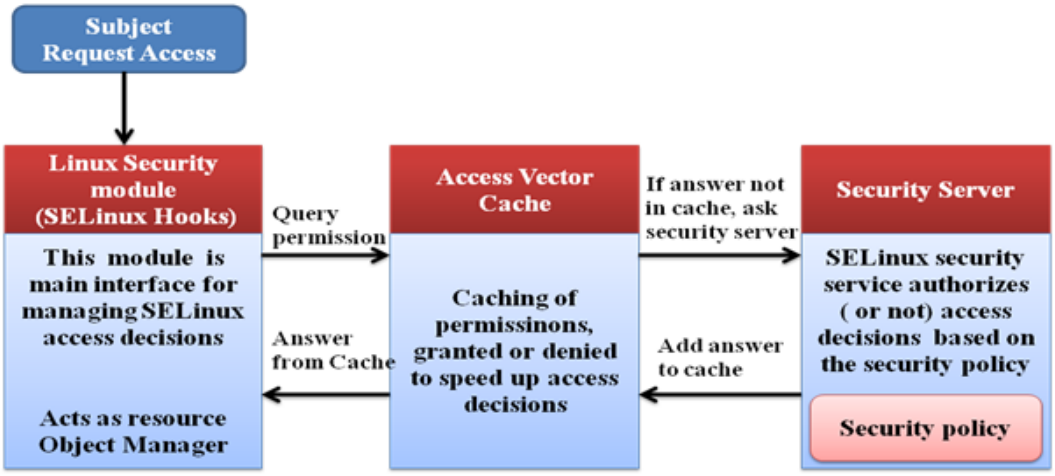
Trong thập niên 90, các chuyên gia của Cục An ninh Quốc gia Mỹ (NSA) phối hợp với tập đoàn Secure Computing Corporation (công ty phát triển loại firewall có sử dụng cơ chế MAC thường được dùng cho các tổ chức quân sự và quốc phòng của Mỹ) xây dựng nên một cơ chế dymanic security policies (chính sách bảo mật động) gọi là Flask được phát triển lên từ dự án Fluke. Sau đó NSA phối hợp với Network Associates và MITRE để đưa cơ chế Flask vào trong Linux, tạo thành SELinux và chính thức công bố SELinux vào tháng 12 năm 2000 [11].

- ***Cơ chế làm việc***

Tăng cường bảo mật trong Linux (SELInux – Security Enhanced Linux) là một cơ chế điều khiển truy nhập hiệu quả được tích hợp vào hệ điều hành Linux từ sau phiên bản 2.4. SELinux có thể được xem là giải pháp bảo mật dữ liệu một cách toàn diện bởi nó áp dụng kết hợp nhiều cơ chế điều kiểm soát truy nhập vào để thực hiện mục đích bảo mật dữ liệu của mình. Có thể coi SELinux là sự tổng hợp của các cơ chế DAC, MAC, RBAC, TE, MLS/MCS.

Trong cơ chế SELinux, mỗi chủ thể (Subject) và đối tượng (Object) sẽ được gán cho một nhãn gọi là Security Context (ngữ cảnh bảo mật). Nhãn này có thể được coi như tấm vé lưu hành cho các Access request (yêu cầu truy nhập). Mỗi nhãn sẽ thể hiện những thông tin về chủ thể hoặc đối tượng đó, bao gồm: User (người dùng), Role (vai trò), Type (kiểu), Security level (cấp độ bảo mật), Security category (phân loại bảo mật).

Security context được coi là tấm vé cho các yêu cầu truy cập, nhưng không phải yêu cầu truy cập nào có tấm vé đó cũng được thực hiện mục đích của mình. Kết quả truy cập là cho phép hay từ chối phụ thuộc vào một thành phần nữa của SELinux chính là Security Server mà nội dung được "chính sách hóa" thông qua Security Policy. Nếu coi hệ thống là một quốc gia, thì Security Server chính là chính phủ và Policy chính là bộ luật để pháp luật được thực thi. Bên trong Policy là những quy định cho phép một Subject được truy cập tới một Object nào đó. Những truy cập chưa được quy định ở Policy thì mặc định được coi là "không hợp pháp".



Hình 1. 3. Mô hình hê thống SELinux

1.5 Kết luận

Trong chương này, thực trạng về vấn đề bảo mật trong doanh nghiệp, nhu cầu cần thiết của một giải pháp bảo mật dữ liệu đã được phân tích, đồng thời giải pháp đang được dùng phổ biến hiện nay mà mỗi công ty hoặc đội ngũ phát triển ưu tiên sử dụngđó là phân quyền và phân cấp trong truy cập cũng được giới thiệu. Cuối cùng, giải pháp sử dụng SELinux mà quá trình nghiên cứu và phân tích cho thấy thấy đây là một giải pháp khả thi và khá hoàn chỉnhsẽ được đề xuất và cung cấp những cái nhìn tổng quan nhất. Để hiện thực hóa ý tưởng này, SELInux đã được áp dụng vào một hệ thống giản lược để chứng minh cho tính khả thi và hiệu quả của giải pháp, những kết quả chứng minh này sẽ được trình bày ở chương 3 của đồ án.

Chương 2. Bảo mật bằng SELinux

Trong thời gian gần đây, đối với những nhà phát triển và những kĩ sư hệ thống yêu thích Linux, coi Linux như là một môi trường hoàn toàn mở cho việc nghiên cứu và tìm tòi thì thuật ngữ "SELinux" đang trở nên rất "nóng". Trên các diễn đàn, hàng loạt những câu hỏi về SELinux, cơ chế bảo mật, lợi ích, thậm chí cả những khó khăn gặp phải khi sử dụng SELinux đã được đăng lên. Điều đó chứng tỏ mặc dù SELinux đã ra đời và được tích hợp vào hệ điều hành được hơn 10 năm, nhưng nó vẫn còn là một khái niệm khá mới mẻ. Quyết định đưa SELinux từ một cơ chế điều khiển truy cập trong máy tính cá nhân trở thành một giải pháp bảo mật trong doanh nghiệp có phần mang tính táo bạo song lại không hề thiếu khả thi. Vậy SELinux là gì và nó có ưu điểm gì nổi bật để giúp cho giải pháp mới này trở nên khả thi như vậy, những nội dung được trình bày trong chương này sẽ trả lời cho những câu hỏi đó.

2.1 Một số khái niệm cơ bản

- ***Subject***

Một chủ thể (Subject) là một thực thể hoạt động đại diện cho một User, một tiến trình, hoặc một ứng dụng gây ra luồng thông tin giữa các đối tượng hoặc thay đổi trạng thái hệ thống [12].

- ***Object***

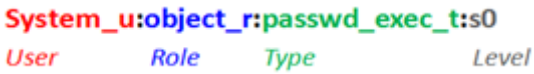
Trong SELinux một đối tượng (Object) là một nguồn tài nguyên như các tập tin, sockets, pipes hoặc giao diện mạng được truy cập thông qua các tiến trình. Các đối tượng được phân loại theo các nguồn tài nguyên mà họ cung cấp với quyền truy cập có liên quan đến mục đích của họ (ví dụ như đọc, tiếp nhận và viết) [13], và chỉ định một ngữ cảnh bảo mật như được mô tả trong các phần sau.

- ***Permissions***

Permission trong SELinux được hiểu là những quyền hạn mà Subject hay Object được cấp khi được khởi tạo.

- ***Security Context***

SELinux đánh dấu tất cả các chủ thể (Subject) và các đối tượng (Object) với một đoạn thông tin được gọi là ngữ cảnh bảo mật (Security context). Một ngữ cảnh bảo mật là cơ chế được sử dụng bởi SELinux để phân loại tài nguyên, chẳng hạn như các tiến trình và các tập tin trên một hệ thống SELinux được kích hoạt. Ngữ cảnh này cho phép SELinux thực thi các quy tắc bảo mật của mình dựa trên chính sách bảo mật được định nghĩa trước. Một ngữ cảnh bảo mật thường được biểu diễn bằng một chuỗi gồm ba hoặc bốn từ. Mỗi từ chỉ định một thành phần khác nhau của ngữ cảnh bảo mật, cụ thể là user (người dùng), role (vai trò), type (loại) , và level (cấp độ) của tập tin hoặc quá trình. Mỗi từ được phân cách bằng dấu hai chấm [14].



Hình 2. 1. Mô tả các thành phần trong một context do SELinux quản lí

- ***SELinux policy***

Chính sách SELinux (SELinux policy) là tập hợp các quy tắc quy định việc thi hành của SELinux. Nó định nghĩa các type (loại) cho các đối tượng tập tin và các miền (domain) cho các tiến trình. Nó sử dụng vai trò (role) để hạn chế các miền (domain) mà tiến trình có thể được truy cập vào, và có định danh người dùng (user identities) để xác định vai trò có thể đạt được. Về bản chất, các loại (type) và các miền (domain) là tương đương, sự khác biệt là type áp dụng đối với các loại đối tượng trong khi các miền áp dụng đối với các tiến trình. Và những câu lệnh cho phép thực thi giữa các domain với type chính là các permission mà những chủ thể có domain đó đạt được.

2.2 Một số cơ chế điều khiển truy nhập trong Linux

2.2.1. Cơ chế điều khiển truy cập tùy quyền

Cơ chế điều khiển truy nhập tùy quyền(Discretionary access control - DAC) là cơ chế điều khiển truy nhập dựa trên chủ sở hữu (Owner) của đối tượng được truy nhập (gọi tắt là đối tượng). Chủ sở hữu của đối tượng thường chính là người đã tạo ra đối tượng đó, có thể quyết định những tiến trình thuộc người dùng hay nhóm nào được quyền truy nhập lên đối tượng đó. Ví dụ như khi người dùng tạo một file, người dùng đó có thể xác định xem những người dùng hay nhóm người dùng nào được quyền truy cập lên file mà người dùng đó vừa tạo ra. Khi một tiến trình yêu cầu quyền truy nhập lên một đối tượng, hệ điều hành sẽ lấy định danh của người dùng (UID) và định danh của nhóm (GID) đang thực hiện tiến trình đó. Sau đó kiểm tra xem người dùng hay nhóm người dùng đó có quyền truy nhập lên đối tượng hay không .

DAC định nghĩa ba quyền truy nhập cho 1 file bao gồm: đọc (read), ghi (write) và thực thi (execute). Trong hệ thống Linux ba quyền này được viết tắt lần lượt bằng ba kí tự r, w và x. Để xem các quyền truy nhập lên từng file có thể dùng lệnh *ls -l* trong cửa sổ dòng lệnh Termina. Có ba cách được dùng để lưu quyền truy nhập lên các đối tượng của hệ thống là dùng ma trận điều khiển truy nhập (Access Control Matrix), danh sách điều khiển truy nhập (Access Control List) và danh sách khả năng (Capabilities List).

Access Control Matrix (ACM) là một ma trận ba chiều trong đó hàng là chủ thể của tiến trình (người dùng hoặc nhóm người dùng, gọi tắt là chủ thể), cột là các đối tượng và giao giữa hàng và cột lưu một tập hợp các quyền mà chủ thể được phép thực hiện với đối tượng đó. Với cách lưu trữ này, số bản ghi sẽ tỉ lệ thuận với các chủ thể và đối tượng. Và do phần lớn các chủ thể không được phép truy cập lên hầu hết các đối tượng nên ma trận này rất thưa thớt. Do đó phương pháp này gây lãng phí bộ nhớ và quá trình truy xuất thông tin tốn thời gian cũng như tài nguyên hệ thống.

Access Control Lists(ACL) là cách thức lưu trữ một bảng gồm các chủ thể và các quyền mà chủ thể đó có thể truy cập lên từng đối tượng. Ví dụ như với một file có

ACL chứa (someone, read) thì someone có quyền đọc file này. Mỗi đối tượng có thể truy cập được sẽ chứa một định danh tới ACL của nó. ACL là phương thức lưu giữ thông tin điều khiển truy nhập mặc định trên các hệ thống UNIX. Phương thức lưu trữ sử dụng Capabilities List cũng tương tự ACL. Tuy nhiên thay vì lưu các chủ thể và quyền của chủ thể lên đối tượng, Capabilities List lưu trữ các đối tượng và quyền mà chủ thể được phép thực hiện trên các đối tượng đó.

Phương pháp kiểm soát truy nhập theo cách thức DAC cho phép điều khiển truy nhập một cách chính xác lên các đối tượng của hệ thống. Phương pháp này đơn giản, hiệu quả, có thể dễ dàng để triển khai mô hình quyền truy nhập tối thiểu (least-privilege model). Nghĩa là nhà quản trị có thể hạn chế chỉ cấp cho chủ thể các quyền tối thiểu nhất họ cần lên từng đối tượng. Tuy nhiên DAC cũng có một số hạn chế nhất định. Đó là tạo lỗ hổng hệ thống cho các Trojan, tức là một tiến trình chạy dưới định danh của một người sẽ có tất cả các quyền mà người dùng đó có mà không thể hạn chế quyền của từng tiến trình một cách hiệu quả. Quá trình bảo trì và phân tích tính bảo mật trong hệ thống DAC rất khó khăn do người dùng tự quản lý các đối tượng của họ. Bên cạnh đó, do DAC không xác định quyền copy nên một tiến trình vẫn có thể lấy được thông tin trong một file mà nó không có quyền được truy cập nếu nó sao chép file đó sang một thư mục khác và đọc thông tin trong file được sao chép.

2.2.2. Cơ chế điều khiển truy cập bắt buộc

Cơ chế điều khiển truy nhập bắt buộc (Mandatory Access Control – MAC) là phương pháp điều khiển truy nhập mà trong đó mỗi tiến trình và đối tượng được gán một tập hợp các trường bảo mật. Khi một tiến trình yêu cầu quyền truy nhập lên một đối tượng, hệ thống sẽ kiểm tra các trường bảo mật này, so sánh với quy định bảo mật được định sẵn trong hệ thống để quyết định cho phép hay từ chối quyền truy nhập của chủ thể. Như vậy khác với DAC dựa trên người dùng, phương pháp truy nhập này dựa trên hệ thống để điều khiển truy nhập. Quy định bảo mật (security policy) được quản lý

tập trung bởi hệ thống, người dùng không thể chỉnh sửa hay thay thế các quy định này. SELinux hỗ trợ hai loại hình của MAC bao gồm:

- **Type Enfocement** - Các tiến trình chạy trong các miền (domains) được xác định bởi loại của nó. Và các thao tác của các tiến trình được điều khiển bởi chính sách của hệ thống. Phương pháp này được sử dụng như một phương pháp điều khiển truy nhập đa năng trong SELinux. Chi tiết về các chức điều khiển truy nhập Type Enfocement sẽ được trình bày trong một mục riêng ở phần sau.
- **Multi-Level Security** - Các đối tượng được phân chia thành các cấp độ bảo mật khác nhau. Và một tiến trình phải có một cấp độ truy cập phù hợp mới được truy cập và chỉnh sửa các đối tượng đó. Phương pháp này đề ra các luật bao gồm "no read up" và "no write down". Nghĩa là chủ thể không thể đọc các đối tượng có mức độ bảo mật cao hơn. Và ghi ra các file có mức độ bảo mật thấp hơn nó. Chi tiết thêm về các luật này sẽ được trình bày trong phần multi-level security.

2.2.3 Phương thức kiểm soát việc truy cập dựa trên cơ sở vai trò

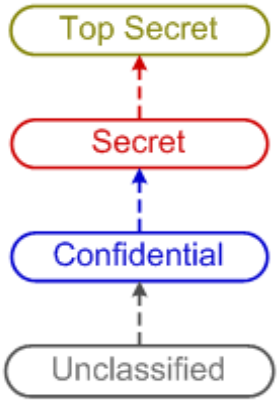
Khái niệm về điều khiển truy nhập dựa trên cơ sở vai trò của người dùng (Role Based Access Control - RBAC)bắt đầu với những hệ thống đa người dùng (multi-user) và đa ứng dụng (multi-application) đã đi tiên phong từ những năm 1970. Nội dung cần chú ý của cơ chế này là các permission được gán cho các vai trò, và những người dùng được gán cho một vai trò thích hợp. Những vai trò được tạo ra cho nhiều vai trò khác nhau trong công việc ở một tổ chức và những người dùng được gán các vai trò dựa trên trách nhiêm và năng lực của họ. Những người dùng có thể dễ dàng được án lại vai trò của mình và những vai trò có thể được cấp những permission mới.

Mục tiêu chính của RBAC là để làm thuận tiện cho việc theo dõi và quản lý an ninh hệ thống. Rất nhiều mô hình điều khiển truy nhập cho những hê thống lớn đã thành công nhờ vào việc quản lý hệ thống dựa trên vai trò. Ví dụ vai trò vận hành có thể truy nhập vào tất cả các tài nguyên của hệ thống nhưng không thể thay đổi các

permission, vai trò người quản trị hệ thống có thể thay đổi permission nhưng không được truy nhập vào các tài nguyên hệ thống, vai trò người kiểm toán viên có thể truy nhập vào các dữ liệu để kiểm tra. Việc sử dụng vai trò trong quản lý cũng được ứng dụng ở hệ thống network, SQL server, và nhiều hệ thống quan trọng khác.

2.2.4 Cơ chế kiểm soát truy cập phân chia cấp độ

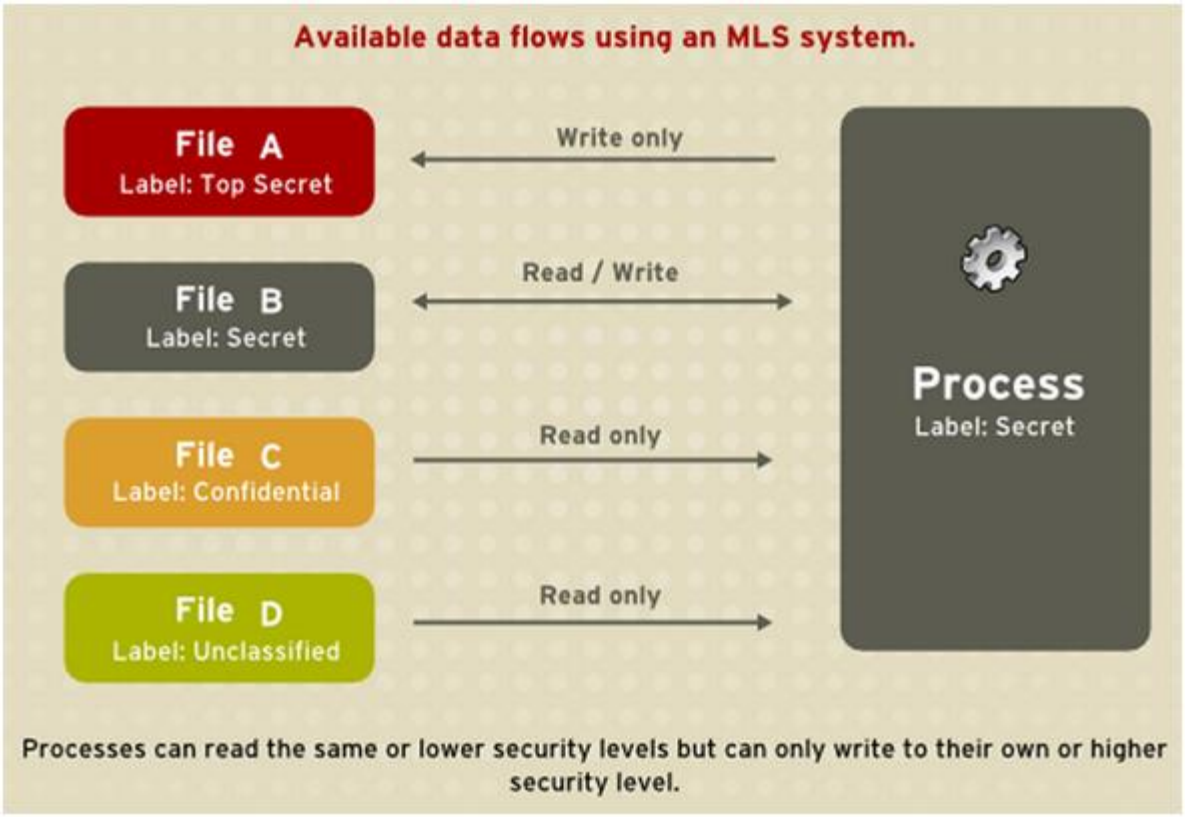
Thuật ngữ "đa cấp độ" (multi-level) phát sinh từ việc phân loại cơ chế bảo mật ra thành 4 cấp độ: không phân loại (inclassified), kín (confidential), bí mật (secret) và tuyệt đối bảo mật (top-secret). Mỗi chủ thể (users hoặc process) phải được cấp phép tương ứng trước khi chúng có thể xem được các thông tin đã phân loại. Những chủ thể mà được cấp phép confidential chỉ được phép xem những tài liệu confidential, chúng được coi là không đáng tin cậy khi xem những thông tin secret hoặc top-secret. Quy định này áp dụng đối với luồng thông tin từ cấp độ thấp tới cấp độ cao hơn và không bao giờ được đảo ngược lại.



Hình 2. 2. Mô hình phân cấp Multi-level secirity (MLS)

Trong các hệ thống, users, máy tính, mạng bảo mật sử dụng các nhãn để chỉ ra cấp độ an ninh. Dữ liệu có thể được truyền/nhận/gửi đi giống như các luồng giữa các cấp độ này. Ví dụ giữa secret và secret, hoặc từ cấp độ thấp hơn tới một cấp độ cao hơn. Điều đó có nghĩa là user ở cấp độ secret có thể chia sẻ dữ liệu với một cấp độ khác và cũng nhận thông tin từ cấp độ khác. Tuy nhiên một chủ thể ở cấp độ cao hơn không thể ghi, cập nhật vào file dữ liệu ở cấp độ thấp hơn. Hạn chế này cũng diễn ra

khi một chủ thể ở cấp độ secret tiến hành việc xem xét các thông tin được phân loại top-secret. Tóm lại có thể tổng kết về việc quản lý dựa trên cơ chế MLS bằng cụm từ "no read up, no write down" (không được đọc thông tin từ các đối tượng có cấp độ cao hơn và không được ghi vào những đối tượng có cấp độ thấp hơn). Cơ chế này được minh họa bởi mô hình 2.3.



Hình 2. 3. Minh họa bảo mật sử dụng mô hình MLS

2.2.5 Cơ chế kiểm soát truy cập phân loại

Kiểm soát truy cập phân loại(Multi-Category Security - MCS) là sự tăng cường cho các hệ thống và hệ điều hành cần tính bảo mật cao. Selinux ứng dụng Multi-Category Security để cho phép users gắn nhãn vơi các category (loại) khác nhau. Những category này được sử dụng để hạn chế một cách chặt chẽ hơn cho cơ chế sơ khai được sử dụng để kiểm soát và điều khiển truy nhập đó là DAC (Discretionary Access Control) và cơ chế Type Enforcement (TE). Cơ chế Multi-Category Security

này cũng có thể hữu dụng khi cần hiển thị hoặc in ra 1 file. Một ví dụ về category là "Company_confidential". Chỉ users có category này mới có thể truy nhập tới những file được gắn cho category này [16].

Thuật ngữ "category" đề cập tới những “loại dữ liệu” không phân cấp và được sử dụng cùng với Multi-Level Security. Trong cơ chế MLS , các Objects và các Subjects được gắn nhãn với các cấp độ an ninh khác nhau. Những cấp độ an ninh này bao gồm những giá trị mức được phân cấp, ví dụ như Top-secret, với không hoặc nhiều category, ví dụ như "Crypt". Các category cung cấp các thành phần bên trong các cấp độ an ninh và thực thi theo nguyên tắc security tương ứng [17].

Multi-Category Security có thể được xem là "sự gia cố" cho MLS . Từ quan điểm kĩ thuật, MCS là sự thay đổi policy, tích hợp thêm những thay đổi để ẩn đi một số kĩ thuật không cần thiết của MLS. Một số thay đổi từ nhân cũng được thực hiện, nhằm mục đích đơn giản hóa để nâng cấp cho MCS hoặc MLS mà không làm thay đổi toàn bộ nhãn của các file hệ thống [18].

MCS gắn nhãn cho users và người quản trị hệ thống một cách trực tiếp, dễ dàng. Nó bao gồm việc cấu hình một tập hợp các category mà đơn giản là các nhãn dạng text, ví dụ như nhãn "Company_Confidential" hoặc 'Medical_Records", và sau đó gán cho users những nhãn này. Người quản trị hệ thống đầu tiên sẽ cấu hình cho các nhãn, sau đó phân cho users như đã được yêu cầu, users có thể sử dụng những nhãn này để thực hiện truy nhập [18].

Tên của các category và ý nghĩa của chúng được thiết lập bởi các quản trị viên của hệ thống, và có thể thiết lập bất cứ category nào được yêu cầu cho một công việc cụ thể. Một hệ thống trong môi trường gia đình có thể chỉ có category là "Private", và được cấu hình để chỉ coi là tin cậy đối với những local users và gán cho họ category

này. Còn trong những môi trường lớn như các tập đoàn, category được sử dụng để xác nhận các tài liệu mật cho các phòng ban nhất định. Category có thể được thiết lập cho các bộ phận nhưng "Tài chính", "Trả lương", "Tiếp thị" hoặc "Nhân sự". Chỉ những users được gán những category này mới có thể truy nhập vào các tài nguyên được gắn những nhãn tương ứng như vậy.

❖ **So sánh MLS và MCS**

Có một số khác nhau cơ bản giữa MCS và MLS [19]:

1. MCS sẽ lờ đi những cấp độ sensitive. Tất cả đều được gắn vúi cùng mức nhạy là "s0", điều này làm cho ý tưởng về mô hình phân cấp sensitivity bị biến mất. Những thiết kế an ninh phân cấp thường ứng dụng không được tốt bên ngoài môi trường quân sự hoặc những môi trường đòi hỏi tính bảo mật cao.
2. MCS cũng loại bỏ đi mô hình an ninh ninh Bell-La Padula (BLP). BLP với những thuộc tính như No-Write-Down mà được thiết kế để ngăn chặn sự rò rỉ thông tin từ những cấp độ an ninh cao tới những cấp độ thấp, thường gân trở ngại và làm ảnh hưởng tới các phần mềm. Trong mô hình BLP cũng không cho phép một root users không được phép Write đến thư mục /tmp cũng như nhiều vấn đều liên quan tới chia sẻ dữ liệu.
3. Multi-Category Security là tùy quyền, tương tự như mô hình chuẩn của unix là DAC.
4. MCS luôn luôn chạy dưới dùng một cấp độ an ninh nình. Cấp độ hiện tại của tất cả các Subjects trong hệ thống là "s0". Khi một Subjects được cấp phép truy nhập tới các category, nó được thực hiện bởi việc thêm ác category vào quyền của Subjects. Hoạt động củ MCS tương tự với cơ chế bổ sung group của Unix: đó là một process có thể truy nhập vào một số nhóm nhưng không được thực thi trong đó. Tương tự, trong MCS, một

process có thể truy nhập vào một tập các category nhưng không được thực thi ở cấp độ an ninh mà có chứa chúng.

Multi-Category Security dử dụng kĩ thuật của MLS nhưng bản thân nó không phải là MLS. Rõ ràng hơn nữa thì MCS không cung cấp bất cứ bảo vệ nào chống lại những phần mềm độc hại hoặc những đe dọa cho hệ thống. Tuy nhiên, cần lưu ý rằng nó đứng trên so với cơ chế Type Enforcemet (TE), cả hai mô hình DAC và TE đều được kiểm tra trước mô hình MCS.

2.2.6 Cơ chế kiểm soát truy cập thực thi theo miền/kiểu

Trong SELinux, tất cả các truy nhập cần phải được cấp phép, tức là không có truy cập nào được xem là mặc định, và cũng không có khái niệm superuser, không giống như quyền root trong Linux. Cách mà một phiên truy cập được cấp phép là định nghĩa type (kiểu/miền) của chủ thể và đối tượng, sau đó sử dụng câu lệnh "allow rule" để cấp phép cho phiên truy cập giữa chủ thể và đối tượng đó. Câu lệnh "allow rule" có 4 thành phần:

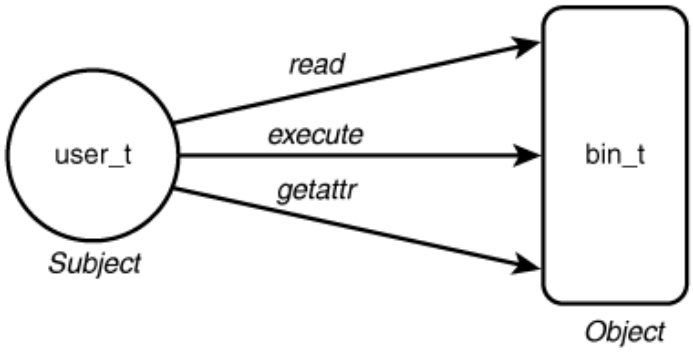
- Miền của chủ thể (Source domain): miền của tiến trình đang tiến hành truy cập.
- Kiểu của đối tượng (Targer type): Kiểu của một file hay một tài nguyên được được truy cập bởi tiến trình nào đó.
- Phân lớp đối tượng (Object class): Phân lớp đối tượng mà một truy cập xác định được cấp phép.
- Quyền truy cập (Permission): Kiểu truy cập mà Subject được phép truy cập tới Object.

Câu lệnh sau là một ví dụ về "allow rule" trong ngôn ngữ policy:

```
allow user_t bin_t : file {read execute getattr};
```

Ví dụ này thể hiện một cú pháp cơ bản trong "allow rule" của cơ chế TE. Trong đó chủ thể có miền là user_t và đối tượng có kiểu bin_t. Định danh "file" là tên của một phân lớp đối tượng được định nghĩa trong policy, và trong trường hợp này, "file"

đại diện cho một tệp tin gốc. Permission được đặt trong cặp dấu ngoặc móc là quyền hợp pháp mà một chủ thể user_t được thực hiện. Câu lệnh trên quy định cho hệ thống hiểu rằng "một tiến trình có miền user_t có thể đọc, thực thi và lấy những thuộc tính về một file đối tượng có kiểu là bin_t".



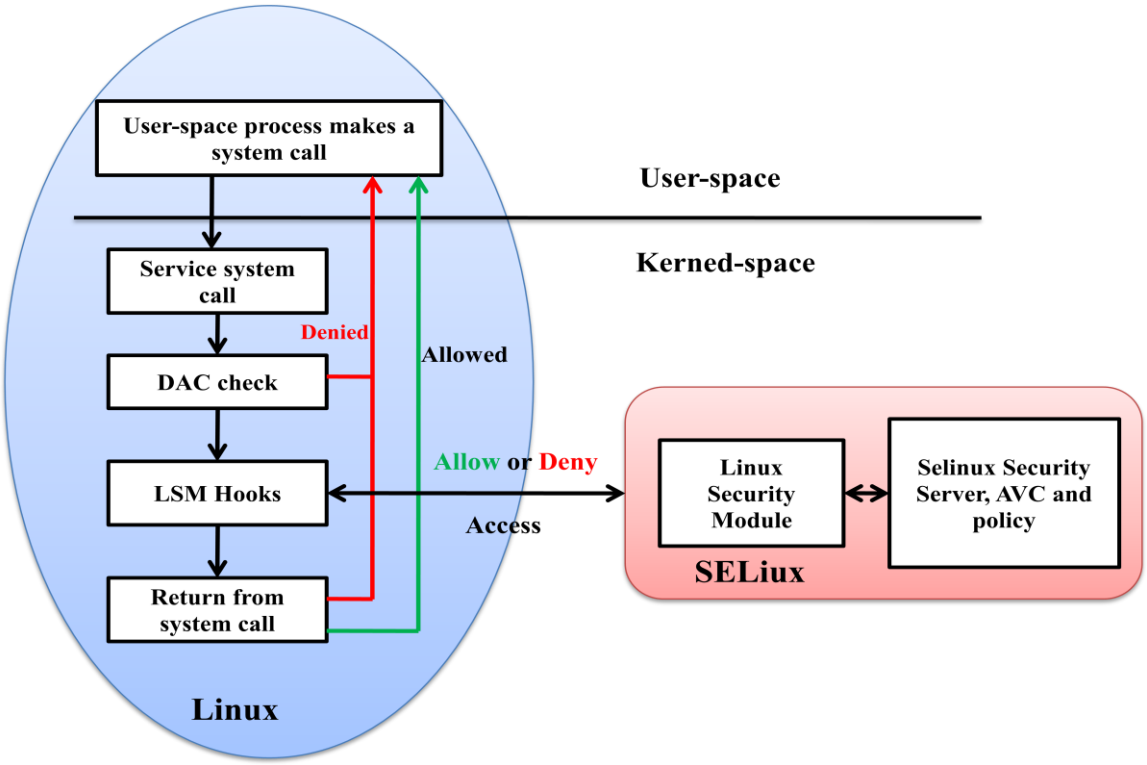
Hình 2. 4. Minh họa permission giữa user_t với bin_t được định nghĩa trong policy

Những permission trong Linux được phân chia rõ ràng và cụ thể hơn trong SELinux. Trong Linux định nghĩa 3 permission cơ bản là read/write/executive (rwx). Trong ví dụ trên, có thêm permission getattr (get attributes), cho phép một process có thể lấy được những thông tin về đối tượng như ngày tháng, thời gian và chế độ DAC đang kiểm soát đối tượng đó. Trong hệ thống Linux chuẩn, một người dùng hoàn toàn có thể xem được những thông tin này của đối tượng, ngay cả khi anh ta không có quyền đọc file đó. Ngoài ra trong SELinux còn rất nhiều permission được định nghĩa để đảm bảo tính chặt chẽ tuyệt đối khi tiến hành kiểm soát truy cập đối với các tiến trình.

Không có quy định nào cho việc đặt tên đối với miền/kiểu của Subject/Object. Nhà phát triển có thể đặt bất cứ tên nào có ý nghĩa và tiện lợi, chỉ cần đúng với quy tắc định danh trong SELinux. Tuy nhiên những miền/kiểu có tên mang tính chất gợi nhớ, ý nghĩa và thống nhất theo một quy tắc chung được khuyến khích để đảm bảo tính mạch lạc và dễ hiểu cho cộng đồng "mở" này có thể tham khảo và kế thừa một cách dễ dàng.

2.3 Mô hình kiểm soát truy cập của hệ điều hành sau khi tích hợp SELinux

Hầu hết các hệ điều hành dùng các phương pháp kiểm soát truy cập (acess control) để xác định một thực thể (người dùng hay chương trình) có thể truy cập vào một tài nguyên nào đó hay không. Các hệ thống dựa trên Linnux thông thường dùng cơ chế *discretionary access control* (DAC) để xác định điều khiển truy cập.

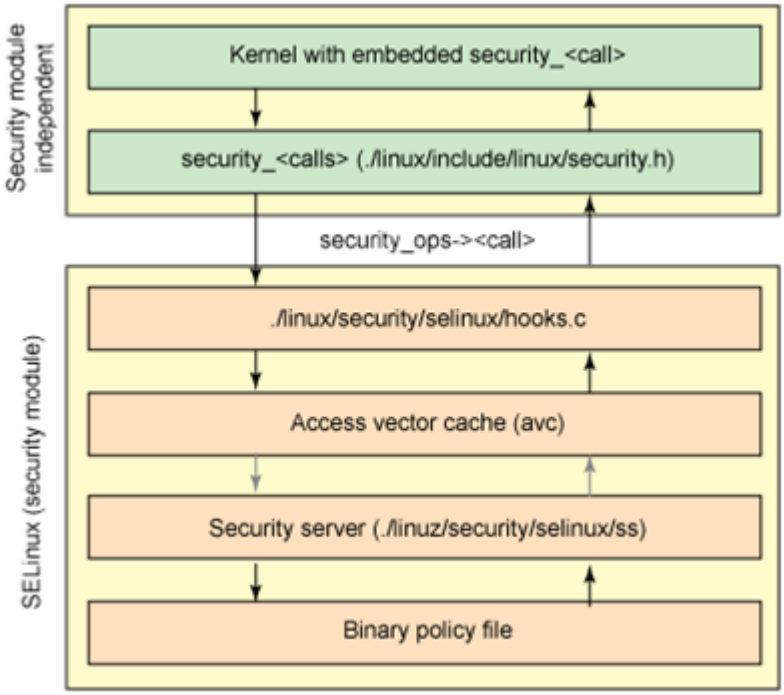


Hình 2. 5. Quá trình kiểm soát truy cập của SELinux

Khi hệ thống Linux được tích hợp thêm SELinux, nếu một tiến trình được cơ chế DAC cho phép truy cập vào tài nguyên thì tiến trình đó tiếp tục được SELinux kiểm tra quyền truy cập thông qua hệ thống các Hook (móc nối) được đặt trong kernel, các Hook này gọi tới các module bảo mật tương ứng để thực hiện chức năng bảo mật. Trong source code của Linux, các Hook này được đặt ở những vị trí nhạy cảm và quan trọng, để thực hiện công việc kết nối sang SELinux. Có thể coi các Hook này như những cây cầu giúp cho quá trình kiểm soát truy cập đang được thực hiện ở cơ chế

DAC nguyên thủy được liên kết với những cơ chế khác trong SELinux. Khi yêu cầu truy cập được gửi sang SELinux thì việc quyết định cho phép hay từ chối truy cập thực hiện dựa vào hệ thống Security Server trong SELinux. Nếu policy trong module bảo mật SELinux cho phép truy cập thì lúc này tiến trình mới được phép truy cập vào tài nguyên hệ thống. Hình 2.5 miêu tả quá trình một Subject (hay process) truy cập vào tài nguyên hệ thống (Object) của hệ điều hành khi có tích hợp SELinux. Chi tiết về cấu trúc và nguyên lý làm việc của SELinux sẽ được trình bày ở các phần tiếp theo trong chương này.

2.4 Cấu trúc hệ thống SELinux



Hình 2. 6. Cấu trúc phân lớp của hệ thống SELinux

2.4.1 Module bảo mật trong Linux

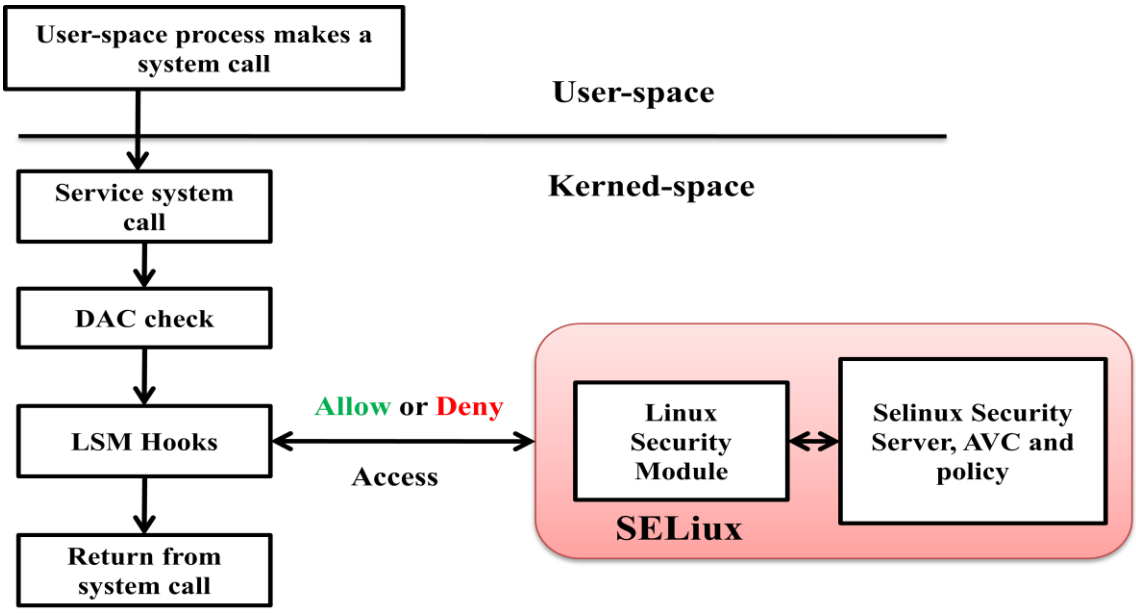
Trong thời kỳ đầu của SELinux, khi vẫn còn là một bộ các bản vá, nó đã đưa ra framework security riêng. Điều này trở thành rắc rối bởi vì nó gắn GNU/Linux vào một kiến trúc điều khiển truy cập duy nhất. Thay vì chấp nhận một cách điều khiển duy

nhất như vậy, Linux kernel thừa kế một framework chung, trong đó tách biệt phần policy và phần thi hành. Giải pháp này được biết đến với tên gọi Linux Security Module (LSM) framework. SELinux lồng vào nhân thông qua LSM framework. LSM cung cấp một framework đa năng dành cho bảo mật, trong đó cho phép các mô hình bảo mật được hiện thực như là những module mà kernel có thể tải lên được.

LSM Framework chủ yếu tập trung vào việc hỗ trợ, cung cấp cơ sở hạ tầng cho các module kiểm soát truy cập, tự nó không thêm bất kỳ security bổ sung nào. Ý tưởng phía sau LSM là cho phép những modules security tích hợp vào trong nhân để có thể tăng cường bảo mật cho cơ chế DAC. LSM cung cấp một bộ các móc (hook). Những hook này thường được đặt sau khi truy cập theo tiêu chuẩn Linux đã kiểm tra, nhưng trước khi tài nguyên thực sự được truy cập bởi kernel với tư cách là lời gọi tới hệ thống. Trong trường hợp không có module bảo mật nào được tải lên thì nó sẽ chứa những hàm dummy (/linux/security/dummy.c). Những hàm này sẽ hiện thực policy theo chuẩn Linux DAC.

Các bản vá LSM hạt nhân thêm trường bảo mật đến cấu trúc dữ liệu hạt nhân và chèn các lời gọi (Hook) tại các điểm đặc biệt trong mã của nhân (code kernel) để thực hiện kiểm tra điều khiển truy cập một module cụ thể. Các Hook LSM được tổ chức thành hai loại: Hook để xử lý các trường bảo mật và Hook để thực hiện kiểm soát truy cập. Khi một nguồn tài nguyên Linux được tạo ra, nhãn bảo mật được gắn liền với nó. Các nhãn được sử dụng để thực thi kiểm soát truy cập bắt buộc với các móc bảo mật. Khi đối tượng bị xóa, nhãn được lấy ra. Hook để xử lý các trường bảo mật được sử dụng để tạo nhãn và loại bỏ nhãn (Ví dụ: alloc_security và free_security trong cấu trúc task_security_ops).Ngoài ra, LSM cũng gắn thêm các hàm để đăng kí và hủy đăng kí các module bảo mật [20].

Trường bảo mật LSM đơn giản chỉ là các con trỏ void*. Để thực hiện bảo mật thông tin tiến trình và chương trình, trường bảo mật được thêm vào cấu trúc task_struct và cấu trúc linux_binrpm. Để bảo mật thông tin filesystem, một trường bảo mật được thêm vào cấu trúc super_block. Để bảo mật thông tin pipe, file, và socket, trường bảo mật được thêm vào cấu trúc inode và cấu trúc file. Để bảo mật thông tin cho packet và network device, trường bảo mật được thêm vào cấu trúc sk_buff và cấu trúc net_device [20]…



Hình 2. 7. Sự Phân nhánh LSM

Một sự phân nhánh của LSM đó là SELinux chỉ được khuyến khích kiểm tra nếu truy cập thành công theo Linux chuẩn. Trong thực tế, điều này không ảnh hưởng tiêu cực trên chính sách điều khiển truy cập bởi vì sự điều khiển truy cập SELinux có thể hạn chế hơn chuẩn Linux DAC và không đè lên quyết định của DAC. Tuy nhiên, LSM có thể ảnh hưởng đến dữ liệu kiểm soát được tập hợp bởi SELinux. Ví dụ, nếu muốn dùng dữ liệu kiểm soát của SELinux để theo dõi tất cả những truy cập bị từ chối, thì trong đa số trường hợp SELinux sẽ không được tham khảo, vì vậy sẽ không thể kiểm

soát được, nếu bảo mật Linux chuẩn từ chối. LSM framework bao hàm những hook được rải rác khắp kernel. Mỗi LSM hook phân bổ cho một hoặc nhiều quyền truy cập cho một hoặc nhiều lớp đối tượng.

❖ **Các móc nối (Hook)**

Mỗi Hook LSM là một hàm con trỏ trong bảng global được gọi là security_ops. Bảng này là một cấu trúc security_operations theo định nghĩa trong include/linux/security.h (thư viện được tạo trong mã nguồn của Linux). Các Hook lại được nhóm lại thành các bộ logic dựa trên các đối tượng hạt nhân (ví dụ : task, inode, sock, file..). Mỗi hàm static inline được định nghĩa một hook. Vì vậy tất cả các cuộc gọi tới hook có thể dễ dàng được biên dịch. Sự phân bố của các lời gọi hook trong code kernel được mô tả bởi dòng "called:" trong tài liệu hướng dẫn cho mỗi hook ở phần đầu của file [20].

Bảng 2. 1. Các Hook LSM

Program execution	Filesystem operations	Inode operations
File operations	Task operations	Netlink messaging
Unix domain networking	Socket operations	XFRM operations
Key Management	IPC operations	Memory Segments
Semaphores	Capability	Sysctl
Syslog	Audit	

Mã nguồn của kernel được hiệu chỉnh để trước khi truy cập vào các đối tượng bên trong, nó sẽ gọi một hook đại diện cho một hàm thi hành. Hàm này hiện thực cho

security policy, nó xác thực hành động có được tiến hành hay không dựa trên những policy được khai báo trước. Các hàm security được chứa trong một cấu trúc các thao tác security bao gồm những thao tác cơ bản cần phải được bảo vệ.

Ví dụ, hàm security_socket_create hook (security_ops->socket_create)kiểm tra các permission trước khi tạo một socket mới và xem xét họ giao thức, loại, giao thức, và socket được tạo trong kernel-space hay user-space [21]. Dưới đây là mã nguồn minh hoạ của file /linux/net/socket.c khi tạo socket.

```
static int __sock_create(int family, int type, int protocol,
        struct socket **res, int kern)
{
        int err;
        struct socket *sock;
        /*
        *      Check protocol is in range
        */
        if (family < 0 || family >= NPROTO)
        return -EAFNOSUPPORT;
        if (type < 0 || type >= SOCK_MAX)
        return -EINVAL;
        err = security_socket_create(family, type, protocol, kern);
        if (err)
        return err;
```

Hàm**security_socket_create** được định nghĩa trong file mã nguồn là /linux/include/linux/security.h. Nó làm trung gian để gọi đến hàm được cài đặt động trong security_ops structure. Xem mã nguồn minh hoạ:

```
static inline int security_socket_create (int family, int type,
        int protocol, int kern)
{
        return security_ops->socket_create(family, type, protocol, kern);
}
```

Hàm trong cấu trúc được cài đặt bởi security module. Bằng cách này, các hook được định nghĩa trong module SELinux mà kernel có thể tải lên được. Mỗi lời gọi SELinux được định nghĩa trong một file chứa các hook hoàn tất việc trung gian giữa hàm của kernel đến lời gọi động cho một security module cụ thể. Mã nguồn minh hoạ từ /linux/security/selinux/hooks.c

```
static int selinux_socket_create(int family, int type,
        int protocol, int kern)
{
        int err = 0;
        struct task_security_struct *tsec;
        if (kern)
        goto out;
        tsec = current->security;
        err = avc_has_perm(tsec->sid, tsec->sid,
        socket_type_to_security_class(family, type,
                protocol), SOCKET__CREATE, NULL);
        out:
        return err;
}
```

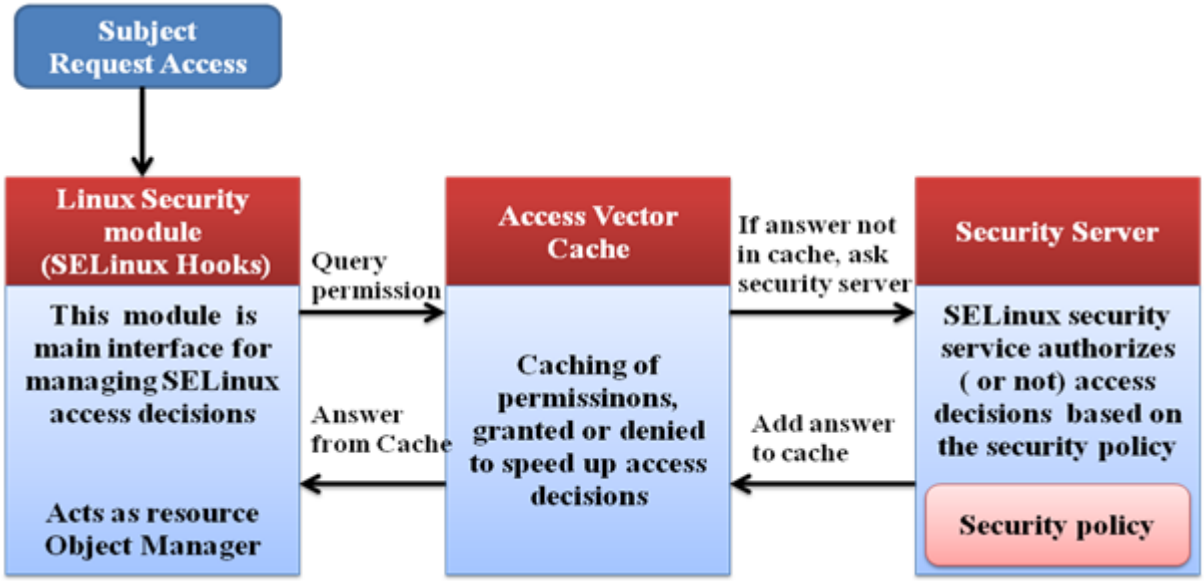
Trọng tâm của đoạn code là lời gọi xác nhận hành động hiện tại của tác vụ hiện tại (được định nghĩa bởi current->security, trong đó current đại diện cho tác vụ hiện tại đang thực thi) có được cho phép không. Lời gọi này bao gồm source security identifier (sid), security class (được tạo thành từ những chi tiết của hành động được yêu cầu), lời gọi socket cụ thể, và tùy chọn cho hỗ trợ kiểm tra dữ liệu. Nếu không tìm thấy quyết định trong AVC, máy chủ security sẽ được gọi để ra quyết định.

Các hàm hook mà security_ops gọi đến được định nghĩa động như là một module kernel có thể tải lên được (thông qua register_security()), còn trong trường hợp không có module nào được load thì nó sẽ chứa những hàm dummy (file /linux/security/dummy.c). Những hàm này sẽ hiện thực policy theo chuẩn Linux DAC. Các hook tồn tại ở tất cả những điểm mà sự can thiệp vào object cần phải được cung cấp để đảm bảo an toàn. Sự can thiệp này gồm có thao tác quản lý (creation, signaling, waiting), nạp chương trình (execve), quản lý hệ thống file (superblock, inode, và filehooks), IPC (message queues, shared memory, và semaphore operations), module hooks (chèn và xoá), và network hooks (gồm có sockets, netlink, network devices, và những giao diện protocol khác). Có thể xem thêm về các loại hook này trong file security.h.

2.4.2 Bộ nhớ lưu trữ kết quả truy cập

Bộ nhớ lưu trữ kết quả truy cập(Access Vector Cache – AVC) là một trong những thành phần chính của hệ thống SELinux, nằm sau khối Linux Security module (làm việc như Resource Object Manager) và nằm trước khối Security Server. AVC đảm nhiệm vai trò như một bộ nhớ lưu trữ các quyết định đã được đưa ra trước đó của hệ thống cho các lượt truy nhập đã diễn ra. Khi yêu cầu truy nhập đi đến AVC thì hệ thống sẽ tìm trong bộ nhớ lịch sử này để kiểm tra, nếu đã tồn tại thì nó sẽ trả về câu trả lời ngay cho yêu cầu truy nhập là cho phép hoặc từ chối, nếu yêu cầu truy nhập chưa từng được lưu trong bộ nhớ lịch sử của AVC thì nó sẽ được gửi tới bộ phận điều khiển

và kiểm soát trung tâm đó là Security Server, sau khi có kết quả thì câu trả lời lại được gửi lại cho AVC và lúc này câu trả lời sẽ được tự động lưu vào bộ nhớ AVC. Hình 2.8 dưới đây là mô hình sơ đồ khối của hệ thống Selinux:



Hình 2. 8. Sơ đồ khối hệ thống SELinux

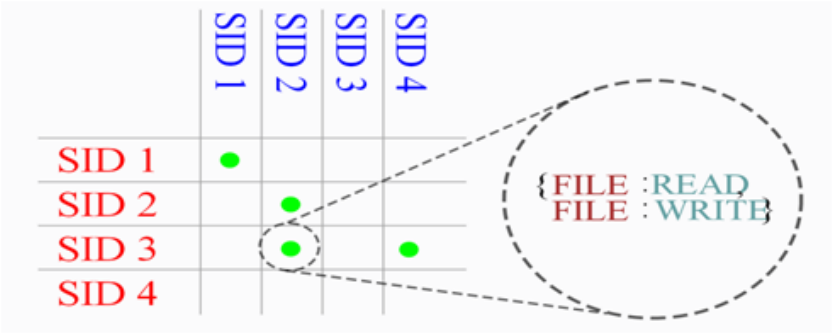
❖ **SID và mô hình quản lý truy cập trong AVC**

SID (Security Identifier) là thuật ngữ được sử dụng nhiều trong hệ thông SELinux nói chung và AVC nói riêng, vậy SID là gì ? Như đã trình bày bên trên, hai khái niệm chính có tương tác và cần quản lý tương tác giữa chúng chính là Objects và Subjects. Objects và Subjects có thể được định nghĩa, mô tả rất phức tạp nhưng cuối cùng chúng vẫn được quy lại thành một khái niệm là SID - một số nguyên không dấu 32 bit (u32) - để dễ dàng cho việc quản lý trong hệ thống cũng như trong các source code cho các nhà phát triển. SID đại diện cho Subjects được kí hiệu là ssid và SID đại diện cho các Objects (hay còn gọi là Target) được biểu diễn bằng tsid [22].

AVC lưu trữ các quyết định (hay là các câu trả lời) cho các yêu cầu truy nhập dưới dạng các cấu trúc avc_decision. Các quyết định này cùng với Subjects, Objects,và một số thông số khác tạp ra 1 cấu trúc avc_entry, được thể hiện trong file nguồn của hệ thống Selinux [23]:

```
struct avc_entry {
        u32             ssid;   // subject SID
        u32             tsid;   // object SID
        u16             tclass; // class
        struct          av_decision  avd;   // contains the set of permissions for
that class
};
```

Hệ thống policy của hệ thông SELinux được mô hình hóa thông qua sơ đồ dạng lưới như hình 2.9.



Hình 2. 9. Mô hình hóa nguyên tắc hoạt động của hệ thống policy trong SELinux [22]

❖ **Khái niệm "action"**

Actions có thể coi là các công việc, hoạt động nà đó được tương tác giữa Subjects và Objects, tức là tương tác giữa các SID với nhau. Một action có thể được mô hình

hóa thành một class và một permission. Một class chứa các đối tượng có chung đặc điểm nào đó và chứa các permission, có tối đa 32 permission trong 1 class. Ví dụ về class như FILE, TCP_SOCKET, X_EVENT. Cho tới thời điểm hiện tại thì hệ thông SELinux có 73 class khác nhau, những class này được định nghĩa trong file selinux//libselinux/include/selinux/flask.h và 1025 permission được định nghĩa trong file selinux//libselinux/include/selinux/AVC_permission.h [23].

Trong hình 2.9, những SID màu xanh đại diện cho các Objects còn các SID màu đỏ đại diện cho các Subjects. Tồn tại một ô giao nhau ứng với 2 SID, ô này là tập hợp các ation mà Subjects và Objects tương ứng có thể thực hiện với nhau. Ví dụ như READ, WRITE.

❖ **Các log ghi lại những action bị từ chối bởi SELinux.**

Trong hệ thống, mỗi kết luận về quyền truy nhập của process tới một Objects đều được lưu trong AVC và được in ra thành các log để phục vụ cho quá trình theo dõi và sửa chữa khắc phục lỗi. Các log này có thể được tìm thấy ở file var/log/audit/audit.log. Mỗi log được ghi lại một các khá dài dòng, chi tiết và gồm những nội dung quan trọng như [23]:

- Process nào bị từ chối?
- Process chạy trong domain nào và bị từ chối khi nào?
- Objects hoặc Subjects nào process đang cố truy cập vào ?
- Subjects/ Objects của file/resource đích ?
- Permission nào bị từ chối ?
- Class của file/resource đích ?
- SID cua process ?
- Số inode của file/resource đích ?
- Điều gì đã xảy ra ?

Có một số chú ý khi theo dõi các log của AVC [23]:

- Không phải log nào cũng báo một process vi phạm quyền truy nhập, việc từ chối truy nhập có thể là do ảnh hưởng của những action khác trong cùng một ứng dụng.
- Việc từ chối các quyền truy nhập có thể diễn ra liên quan với nhau, việc từ chối process này có thể dẫn tới việc từ chối 1 process khác.
- Có thể có nhiều chuỗi quyết định từ chối process liên tiếp nhau và được quy định sẵn trong Kernel.
- Vì những lý do trên nên không phải bất cứ một log nào cũng được in ra mà nó có thể bị ẩn đi.

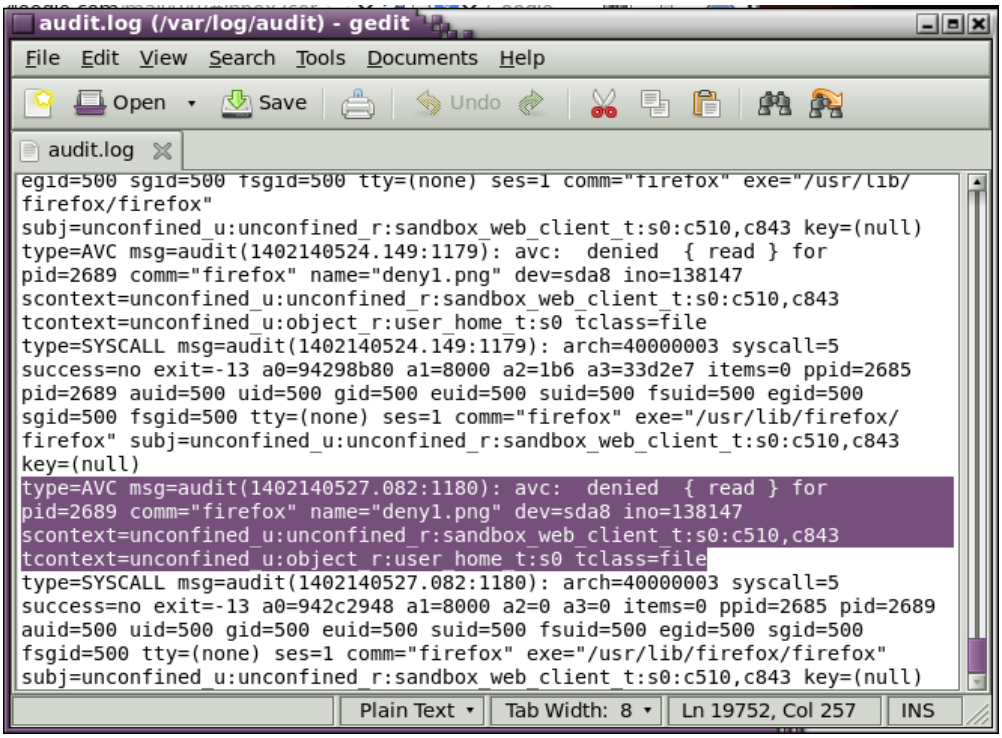
Khi hệ thông SELinux được kích hoạt thì dù nó ở chế độ nào (Enforce hay Permissive) thì hệ thống security cũng sẽ sinh ra các log. Còn một process bị từ chối truy nhập hay không thì tùy thuộc vào từng chế độ, cụ thể là nếu ở chế độ Permissive thì hệ thông SELinux làm việc không khác gì hệ thống Linux bình thường, chỉ có các log được in ra trong file audit.log (file var/log/audit/audit.log) . Còn trong chế độ Enforcing thì Selinux kiểm soát hệ thống một cách nghiêm ngặt [24].

AVC là một khối nằm trong hệ thông SELinux, với mục đích ghi lại nhật kí quản lý và kiểm soát hệ thống làm việc, nó lưu sẵn trong bộ nhớ của mình những quyết định này. Do đó giúp tăng hiệu năng của hệ thống, giảm được những thao tác lặp lại không cần thiết và đưa ra quyết định cho phép hoặc từ chối đối với một process một cách nhanh nhất. Hình 2.10là một ví dụ về AVC log. Bảng 2.2 giải thích các tham số được thể hiện trong AVC log.

2.4.3 Security Server

Security Server là khối cuối cùng trong bộ máy kiểm soát truy nhập Selinux. Nó sẽ đưa ra quyết định cuối cùng về việc một Subjects có được quyền truy nhập vào một

Objects không. Quyết định này dựa trên những quy định của Security Policy. Cho tới thời điểm hiện tại, Security Server được nhúng trong nhân với policy được biên dịch và được chèn vào hệ thống từ userspace thông qua nhiều hàm nằm trong thư viện libselinux [25].



Hình 2. 10. Ví dụ AVC log

❖ **Security Policy trong hệ thông SELinux**

Security Policy đúng như tên gọi của nó là những chính sách an ninh quy định việc thi hành cua hệ thông SELinux. Nếu coi SELinux là một quốc gia thì Security Policy chính là hiến pháp, nó quy định việc một Subjects có được quyền truy nhập vào Objects hay không. Có thể chia làm hai loại policy tùy theo mức độ an ninh toàn mà nó điều khiển hệ thông SELinux.

- Loại thứ nhất là Targeted policy: Trong loại này thì tất cả các subject và object đều chạy dưới cùng 1 domain là unconfined_t (không bị hạn chế), trừ

những tiến trình nền xác định đã có domain riêng của nó. Với cơ chế policy này thì SELinux làm việc tương tự cơ chế DAC của Linux thông thường, không đảm bảo an toàn cho hệ thống.

- Loại thứ hai là Strict Policy, trái ngược lại với Targeted Policy. Mỗi subject và object có một domain riêng của nó, do đó chịu sự quản lý và giới hạn chặt chẽ bởi policy.

Để kiểm tra các thông tin về policy trong Selinux có thể dùng lệnh seinfo trong Terminal của linux:

```
[Nhan@localhost ~]$ seinfo

Statistics for policy file: /etc/selinux/targeted/policy/policy.24
Policy Version & Type: v.24 (binary, mls)

   Classes:            81    Permissions:       235
   Sensitivities:       1    Categories:       1024
   Types:            3693    Attributes:        328
   Users:               9    Roles:              12
   Booleans:          217    Cond. Expr.:       257
   Allow:          296152    Neverallow:          0
   Auditallow:        127    Dontaudit:      229830
   Type_trans:      34472    Type_change:        38
   Type_member:        48    Role allow:         19
   Role_trans:        312    Range_trans:      4581
   Constraints:        90    Validatetrans:       0
   Initial SIDs:       27    Fs_use:             23
   Genfscon:           83    Portcon:           446
   Netifcon:            0    Nodecon:             0
   Permissives:        75    Polcap:              2
```

Hình 2. 11. Xem thông tin về phiên bản SELinux đang sử dụng

Dòng đầu tiên là đường dẫn tới file có tên policy.X, với X là số nguyên chỉ phiên bản của policy đang được dùng trong hệ thống. Trong file nà chứa dạng binary của policy sau khi được biên dịch và hệ thống sẽ làm việc thông qua những quy định được mã hóa nằm trong file này. Ngoài ra còn các thông tin được cung cấp với lện seinfo như user, role, target, permission…

Các policy có thể được xây dựng dưới dạng các module, mỗi module thực hiện quản lý một process. Một module policy được tạo ra thông qua 1 file bắt buộc là file policy_name.te, 2 file khác (optional) có cùng tên nhưng khác đuôi định dạng: policy_name.if, policy_name.fc. Trong đó [25]:

- ***file policy_name.te*** (type enforcement) : chứa những khai báo và chính sách đóng đối với cùng 1 domain (personal or local) .
- ***file policy_name.if*** (interface) : chứa những khai báo và chính sách mở, có thể được chia sẻ giữa các domain muốn tương tác với nhau (shared).
- ***file policy_name.fc*** (file context) : quy định context cho 1 file trong hệ thống. File này liên quan tới hoặc được điều khiển bởi policy đang được build. VD trong file clock.fc thì sẽ định nghĩa policy cho file /etc/adjtime và file /sbin/hwclock.

❖ *VD: localpolicy.te*

```
policy_module(localpolicy, 1.0)
gen_tunable(xdm_sysadm_login, false)
gen_require(`
        type user_t;
        type var_log_t;
       ')
allow user_t var_log_t:dir { getattr search open read };
```

Trong ví dụ trên:

- policy_module (localpolicy, 1.0): module definition, tên và version của policy
- gen_tunable(xdm_sysadm_login, false) : tạo ra một cờ boolean và gán giá trị cho nó.
- gen_require: định nghĩa những class và các access bên trong policy
- Khai báo: khai báo cho Subject và Object (type object_t)
- rules allow …..: khai báo những rule cần thiết ứng với mỗi domain
- các macro: dùng để định nghĩa 1 một tiến trình nền hoặc thực hiện một công việc nào đó.
- Block: Kí tự '…' dùng để đánh dấu bắt đầu và kết thúc của 1 block

❖ Các bước tạo và biên dịch một policy module:

- B1. Tạo 3 file kể trên
- B2. Từ Terminal, tạo file policy_name.pp từ file policy_name.te
- B3. Cài đặt file .pp (policy package) bên trên, kiểm tra bằng lệnh semodule -l đểxem poicy đã được tích hợp vào file policy thực thi của hệ thống

Bảng 2. 2. Các tham số trong một đoạn log của AVC

Log part	Name	Description
Type=AVC	Log type	Thông báo cho người dùng về loại log được tạo ra, trong trường hợp này loại log là AVC
msg=audit(1363289005.532:184)	Timestamp	Nhãn time tính theo giây từ ngày 1.1.1970. Nhãn này có thể được định dạng lại cho dễ theo dõi bằng lệnh date –d @ rồiđến con số tương ứng trong nhãn này. VD: user$ date –d @1363292159.532 Thu Mar 14 21:15:59 CET 2013
avc	Long type (again)	Loại log được thông báo cho người dùng, ởđây là loại avc
Denied	State(if enforced)	Thông báo SELinuxđã từ chối (Denied) hay cấp phép (granted) cho yêu cầu truy nhập. Chúý là nếu SELinuxở trong chếđộ Permissive thì nó vẫn thông báo log như trong chế độEnforcing mặc dù mọi yêu cầuđềuđược cho phép.

{read}	Permission	Permission được yêu cầu/ thực hiện. Trong ví dụ này là yêu cầu đọc (read). Đôi khi Permission có thẻ là một tập, ví dụ như{readwrite} nhưng trong hầu hết các trường hợp thì chỉ có một Permission được yêu cầu
for pid =29199	Permission	ID của process yêu cầu
comm=“Trace”	Process CMD	Lệnh process (thường giới hạn là 15 kí tự) giúp cho cácUser xácđịnh process nào trong trường hợp nóđang thực hiện (một số PID chỉ được sử dụng khi processđó vẫn đang chạy).
name= “online”	Taget name	Tên của Object (hay Target), trong trường hợp này là tên file
dev= “sysfs”	Device	Device trong đó có chứa Object, trong trường hợp một file hoặc file hệ thống
ino=30	inode number	số inode của file đích. Trong trường hợp này, vìđã biết nó nằm trong file hệ thống sysfs (trong /sys), chúng ta có thể tìmđược file này sử dụng lệnh find: user$ find /sys –xdev –inum 30 /sys/devices/system/cpu/online
scontext=staff_u:staff_r:googletalk_plugin_t	Source context	Security context của process. (chính là một domain)
tcontext=system_u:object_r:sysfs_t	Target context	Security context của target (trong ví dụ này là một file).
tclass=file	Target class	Class của target

2.5 Nguyên lý bảo mật của SELinux

2.5.1 Các ngữ cảnh bảo mật.

Trong SELinux, một ngữ cảnh bảo mật (security context) được biểu diễn bằng một chuỗi có chiều dài thay đổi để xác định các SELinux người dùng (user), vai trò của chúng (role), một loại định danh (type identifier) và một dải bảo mật tùy chọn MLS/MSC hoặc cấp độ có định dạng như sau:user:role:type[:range] [26]. Bảng 2.3 mô tả các thành phần của một security context trong SELinux.

Bảng 2. 3. Chức năng của các thành phần trong một Security context

user	Định danh của người sử dụng SELinux. Định danh này có thể được liên kết tới một hoặc nhiều vai trò (role) mà user SElinux được phép sử dụng.
role	Vai trò của SELinux. Nó có thể được liên kết tới một hoặc nhiều loại (type) user SELinux được phép truy cập.
type	- Khi một loại (type) được liên kết với một tiến trình (process), nó sẽ định nghĩa những process (hoặc domains) SElinux user (Subject) được phép truy cập. - Khi một loại (type) được liên kết với một đối tượng (Object),nó định nghĩa những quyền truy cập của SElinux user đến Object đó
range	Đây là trường cũng có thể biết đến như một cấp độ (level) và chỉ xuất hiện nếu policy hỗ trợ MCS hoặc MLS. Mục này bao gồm: -Mức độ bảo mật duy nhất có chứa một mức độ nhạy cảm (sentivity level) và có hoặc không các danh mục (VD: s0, s1:c0, s7:c10.c15) -Một [range] bao gồm hai mức bảo mật (cao và thấp) cách nhau bởi dấu gạch ngang (VD: s0 - s15:c0.c1023)

Các mã kiểu chuỗi nhận dạng giữa các thành phần được định nghĩa trong các phần chính sách (policy) của SELinux (và sẽ được thảo luận kĩ hơn ở phần sau). Thông thường để hiện thị các ngữ cảnh bảo mật của các đối tượng và chủ thể trong Selinux, thường thêm đuôi –Z vào phần các câu lệnh như:

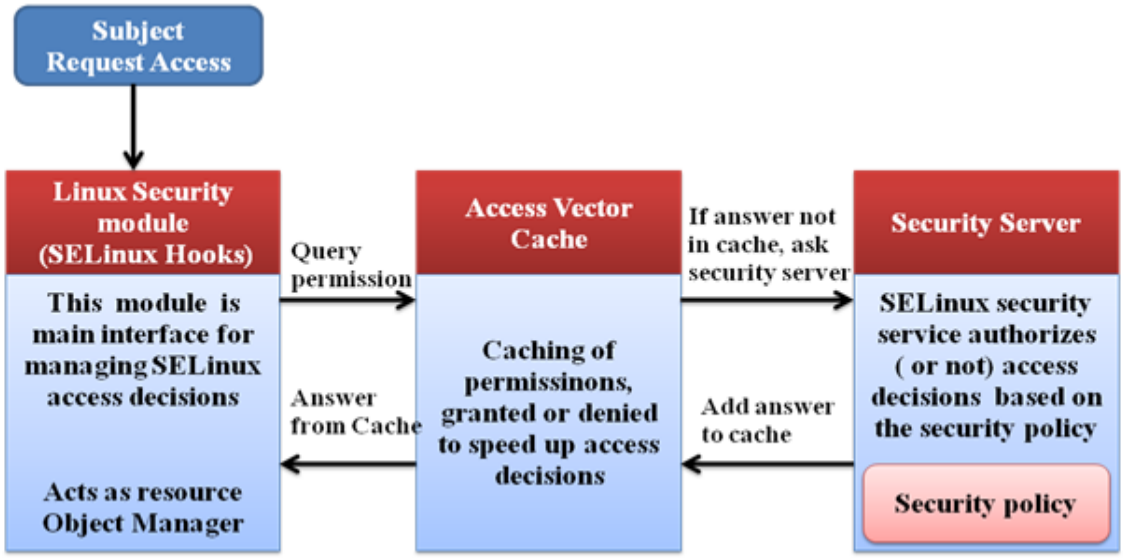
Ls –Z : hiển thị security context của các đối tượng file hệ thống

Ps –Z : hiển thị security context của các tiến trình trong hệ thống

Id –Z : hiển thị cho user, role và type của user hiện thời.

2.5.2 Bảo mật dựa trên kiểm soát truy cập

SELinux tạo khả năng kiểm soát truy cập một cách chặt chẽ đối với tất cả tài nguyên trong kenel.



Hình 2. 12. Các thành phần chính của hệ thống SELinux

Hình 2.12 cho thấy một sơ đồ mức cao của các thành phần trong SELinux,chúng thực thi quản lý các chính sách và gồm những thành phần sau đây [27]:

- Từ không gian người dùng (User space), một chủ thể (Subject) sẽ tạo ra yêu cầu truy cập (Access request) và gửi tới nhân của hệ thống (Kernel space), tới hệ thống SELinux.
- Linux Security module được cấu tạo bởi các hàm móc nối (Hook) sẽ biết các yêu cầu cần gửi đến đối tượng (Object) nào (Chẳng hạn như một tập tin) và có những quyền gì.
- Access Vector Cache (AVC) là nơi lưu trữ các quyết định truy cập trước đó. Khi yêu cầu truy cập được gửi tới đây, AVC sẽ lục tìm trong bộ nhớ của mình để đưa ra quyết định ngay (nếu đã có) hoặc gửi tới những khâu tiếp theo.
- Một máy chủ bảo mật (Security Server) đưa ra các quyết định liên quan đến việc có cho phép Subject tác động đến các Object, dựa trên các quy tắc chính sách bảo mật.
- Một tập các chính sách bảo mật (Security policy) mô tả các quy tắc, sử dụng ngôn ngữ bảo mật SELinux.

Hình 2.13 mô tả chi tiết cấu trúc SELinux và quá trình hoạt động của hệ thống khi có một Subject yêu cầu truy cập đến một Object trong tài nguyên hệ thống. Ngoài ra hình cũng chỉ ra cách can thiệp vào policy bảo mật trong hệ thống:

- Đầu tiên khi một yêu cầu truy cập được gửi từ User space tới Kernal space thì cơ chế DAC của Linux sẽ kiểm tra xem Subject có được truy cập đến Object hay không. Nếu DAC kiểm tra và thấy rằng Object không có quyền truy cậpđến Subject, lúc này quá trình truy nhập của Subject bị từ chối (access denied).

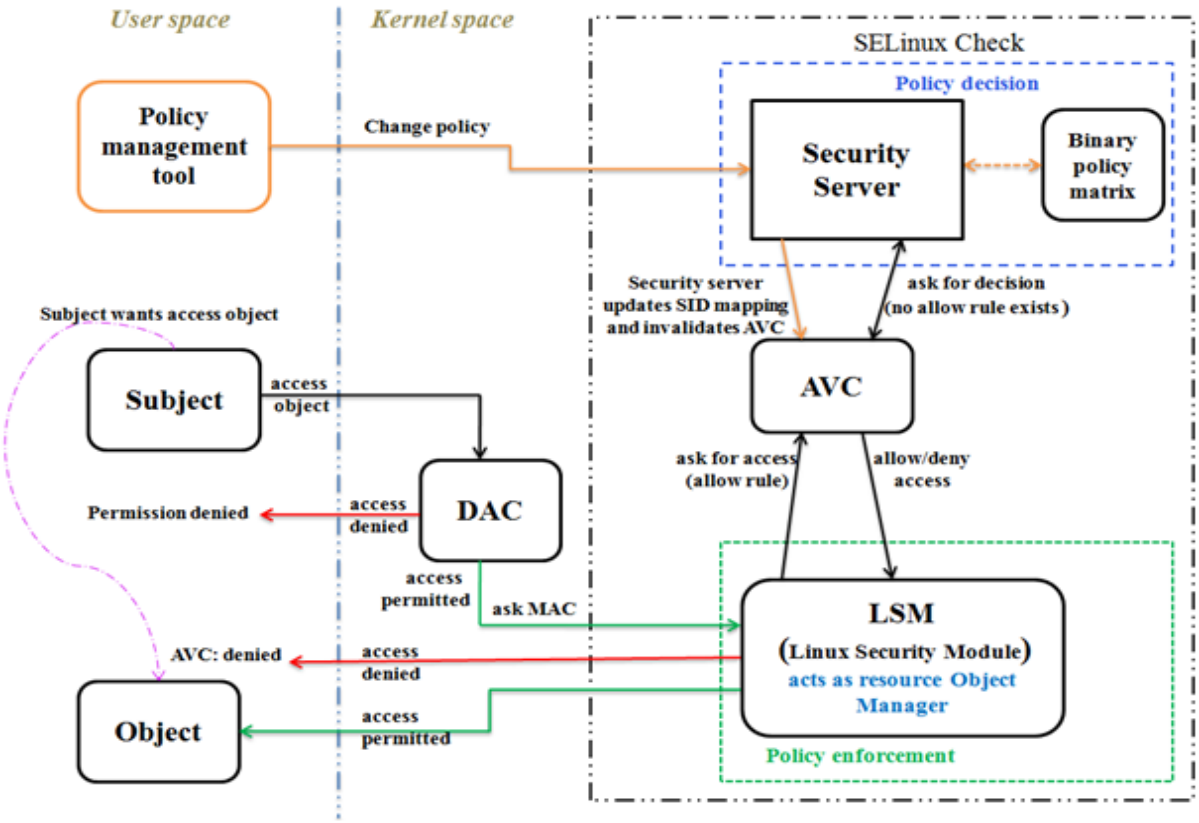
- Nếu DAC kiểm tra và cho phép Subject truy cập, lúc này các hàm **security_call** trong kernel sẽ được gọi để liên kết với LSM hook tương ứng với các module bảo mật của SELinux.

- LSM sẽ thông qua AVC để tìm quyết định truy cập để xác định xem yêu cầu tương tự đã có trước đó hay chưa thông qua hàm **avc_has_perm ()**. Nếu đã có, AVC

sẽ trả kết quả về cho LSM để LSM quyết định có cho Subject truy cập đến Object hay không.

- Nếu trong AVC chưa có quyết định nào liên quan đến Subject và Object, lúc này AVC sẽ gọi tiếp đến Security Server thông qua hàm **security_compute_av().** Security Server sẽ dựa vào các tham số được truyền đến trong hàm **security_compute_av()** và module policy để tính toán và đưa ra kết quả chả về cho AVC. AVC sẽ trả kết quả về cho LSM đồng thời AVC cũng lưu lại kết quả vừa nhận được.

- Cuối cùng LSM dựa vào kết quả nhận được từ AVC sẽ đưa ra quyết định cho phép hay từ chối truy cập.



Hình 2. 13. Kiến trúc chi tiết hệ thống SELinux

Mọi quyết địnhđều dựa trên module policy được nhà phát triển viết ra vàđược tích hợp sẵn trong SELinux. Bằng một số công cụ (Policy Manager Tool) các nhà phát triển có thể chỉnh sửa, xóa policy.

2.5.3 So sánh giữa Linux trước và sau khi tích hợp SeLinux

Trong Linux chuẩn, các thuộc tính quản lý truy cập của 1 chủ thể chính là user và group ID kết hợp với tất cả tiến trình thông qua cấu trúc tiến trình trong hệ thống. Các thuộc tính này được bảo vệ bởi hệ thống và chỉ được thay đổi thông qua các công cụ quản trị như tiến trình login và chương trình setuid. Đối với các đối tượng như file, inode (index node là một khái niệm để lưu thông tin cơ bản như file type, permission, Owner,… của đối tượng) chứa một tập các bit về chế độ truy cập, ID của group và file user. Người ta dùng tập kết hợp của 3 bit read/write/excute để quản lý quyền truy cập.

Trong SELinux, các thuộc tính quản lý truy cập luôn luôn gồm 3 phần (user, role, type). Tất cả các đối tượng và chủ thể trong hệ điều hành đều có một ngữ cảnh truy cập kết hợp giữa 3 thành phần này. Trong khi bản Linux chuẩn dùng ID của tiến trình dành cho từng user, group; chế độ truy cập của file và các ID cho file để cấp quyền truy cập tài nguyên hay ko, bản SeLinux dùng các ngữ cảnh bảo mật của một tiến trình và các đối tượng để cấp quyền truy cập. Những so sánh này được trình bày trong Bảng 2.4.Trong SeLinux, mọi việc truy cập đều phải được cấp quyền một cách rõ ràng thông qua cơ chế TE như đã trình bày bên trên, không có khái niệm cho "truy cập theo mặc định" không quan tâm tới user, group ID nào.

SELinux kiểm soát quá trình truy cập một cách chặt chẽ dựa trên những cơ chế như DAC, MAC, RBAC, TE, MLS. Do đó tránh được các nguy cơ như:

- Quyền hạn bị rơi vào tay người dùng có mục đích xấu, do trong SELinux thì chỉ người quản trị hệ thống mới có quyền hạn tối đa, còn những người dùng khác, ngay cả chủ sở hữu cũng không có được quyền đó. Tất cả đều chịu sự chi phối của Administrator.

- Ngăn chặn các ứng dụng truy cập trái phép vì cơ chế SELinux có thể hạn chế tối thiểu quyền truy cập của các ứng dụng đó. Chỉ cấp đủ quyền cho nó thực hiện công việc của mình.
- Tránh nguy cơ truy cập leo thang nhờ vào hệ thống các cơ chế được kết hợp chặt chẽ.
- Điều khiển, kiểm soát các tiến trình để thực hiện các công việc mà lẽ ra phải sử dụng quyền root. Tránh mối đe dọa thay đổi cấu hình bảo mật của hê thống từ người dùng có mục đích xấu.

Bảng 2. 4. Linux trước và sau khi tích hợp SELinux

Đặc điểm	Bản Linux ban đầu	Sau khi tích hợp SELinux
Các thuộc tính bảo mật của tiến trình	Các user và group ID	Ngữ cảnh bảo mật
Các thuộc tính bảo mật cho đối tượng	chế độ truy cập và các ID của user và group trên file	Ngữ cảnh bảo mật
Các điểm căn bản cho quản lý truy cập	Việc xử lý các ID của user/group và chế độ truy cập dựa vào ID của user/group trên file đó.	Quyền được xem xét giữa tiến trình và đối tượng thông qua các policy.

2.6 Vai trò của SELinux đối với việc bảo mật dữ liệu.

Đối với bảo mật dữ liệu, SELinux không cung cấp một mật khẩu hoặc một phân chia về quyền hạn như DAC. Tất cả đều được bảo mật thông qua kiểm soát truy cập. Cho tới nay, SELinux trở nên tối ưu và quan trọng bởi những lợi ích mà nó mang lại:

- SELinux gắn cho mỗi process và mỗi tài nguyên một nhãn an ninh. Nhãn này đại diện cho process và tài nguyên đó. SELinux lợi thế hơn so với cơ chế DAC thông thường là ở cách mà nó kiểm soát tới từng đối tượng thông qua các policy.
- Chính nhờ cơ chế kiểm soát truy cập một cách chặt chẽ mà SELinux hạn chế được các nguy cơ về truy cập trái phép vào các tài nguyên hệ thống, các tài nguyên chung cần quản lí mà không phải bất cứ người dùng nào cũng có quyền truy cập.
- SELinux kiểm soát các process, do đó có thể ngăn chặn kịp thời các tiến trình có mục đích xấu do các ứng dụng hoặc người dùng tạo ra.
- SELinux có thể hạn chế tối thiểu quyền hạn của các ứng dụng, ngăn chặn các nguy cơ về rò rỉ thông tin.

Với những lợi thế trên, SELinux được sử dụng để bảo mật dữ liệu một cách hiệu quả trên máy tính cá nhân cho nhiều người dùng khác nhau. Trong đề tài này, SELinux sẽ được phát triển thành ứng dụng bảo mật trong doanh nghiệp.

2.7 Kết luận

Như vậy trong chương này, những cơ sở lý thuyết căn bản về SELinux đã được trình bày một cách trọn vẹn trong chương này. SELinux được xem như một cơ chế bảo mật và điều khiển truy cập được tích hợp vào hệ thống, khi làm việc, nó tiến hành kiểm soát sau khi các yêu cầu truy cập đã đi qua và thỏa mãn cơ chế DAC. Để yêu cầu đến được SELinux cần có những hàm móc nối (hook) được đặt ngay trong mã nguồn của Linux, tại những vị trí được xem là nhạy cảm. SELinux chia làm nhiều module, mỗi khối thực hiện mội chức năng và kết hợp với nhau, giúp hệ thống làm việc hiệu quả với một hiệu năng tốt nhất. Để kiểm soát được các chủ thể và đối tượng, đồng thời kiểm soát quá trình tương tác giữa chúng, SELinux gán cho mỗi chủ thể và đối tượng những nhãn được gọi là Security context, và thông qua những chính sách bảo mật (Security policy) để quy định về quyền hạn của các chủ thể và đối tượng. Các nhà phát triển, thậm chí những người dùng yêu thích công nghệ hoàn toàn có thể tự biên dịch và chèn

một policy mới vào nhân để giúp hệ thống có thể làm việc theo ý mình. SELinux làm việc chặt chẽ, hiệu quả nhưng lại linh hoạt. Từ khi ra đời, SELinux đã được ứng dụng vào các mục đích bảo mật, nhằm tạo ra một cơ chế bảo mật tối ưu cho các tổ chức có nhu cầu bảo mật cao như các doanh nghiệp, các cơ quan tình báo, quân sự. SELinux đã mở ra một giải pháp mới cho bảo mật dữ liệu trong các cơ quan, tổ chức thay vì dùng những cơ chế thông thường và đơn lẻ như MLS, RBAC hoặc MAC hiện nay. Thấy được những tính năng vượt trội này của SELinux, giải pháp ứng dụng vào việc đáp ứng nhu cầu bảo mật trong các doanh nghiệp đã dược đề xuất, giải pháp này sẽ được trình bày cụ thể trong chương tiếp theo của đồ án.

Chương 3.Triển khai hệ thống bảo mật trong doanh nghiệp

Trong tình hình kinh tế thị trường ngày nay, các doanh nghiệp cạnh tranh nhau một cách khốc liệt để tồn tại và phát triển. Trong đó không thể loại trừ các biện pháp, thủ đoạn để có được những cơ hội, đối tác, khách hàng và bí quyết sản xuất, kinh doanh của đối phương. Và một lần nữa, vấn đề bảo mật thông tin và dữ liệu trong doanh nghiệp lại được đưa ra bàn bạc với mong muốn tìm ra một giải pháp tối ưu. Trong các công ty, các văn phòng, các tổ chức có những dữ liệu nhạy cảm mà dù có là bí mật kinh doanh hay không thì vẫn luôn được yêu cầu phải giữ kín. Đặc biệt trong thời buổi các trang web xã hội như facebook, google plus, zing… đang có sức ảnh hưởng cực kì lớn đối với mỗi người. Một bức ảnh chụp văn phòng làm việc, một đoạn video clip quay lại không khí vui vẻ của một buổi sinh hoạt văn nghệ nhưng chưa được sự đồng ý mà một cá nhân, nhân viên nào đó tự tiện đưa lên các trang web xã hội cũng có thể được coi là vi phạm chính sách bảo mật của công ty. Trong chương này, giải pháp bảo mật bằng SELinux sẽ được trình bày. Những kết quả khi áp dụng SELinux cũng sẽ được thể hiện để chứng minh cho tính khả thi của giải pháp mới này.

3.1 Thiết kế giải pháp

Trước phiên bản 2.4, Linux đã được tích hợp hai cơ chế bảo mật là DAC và MAC. Và cho tới ngày nay, trong những môi trường làm việc có quy mô nhỏ, người ta vẫn sử dụng và hài lòng với hai cơ chế này. DAC – cơ chế điều khiển truy nhập tùy quyền – đúng như tên gọi của nó, khả năng truy cập tùy thuộc vào việc User đó được gán cho những quyền nào, được phép thực thi những công việc gì. Cho tới khi MAC – điều khiển truy nhập bắt buộc - thay thế DAC thì quyền hạn cao nhất không còn nằm trong mỗi Owner (chủ nhân đối với mỗi tài nguyên) nữa mà Administrator (người quản trị viên) có quyền kiểm soát toàn bộ hệ thống. Nguyên lí làm việc chính của MAC là mỗi tệp tin, tài nguyên cũng như mỗi User, mỗi tiến trình sẽ được gán cho một security label (nhãn an ninh) và quyền hạn của họ phụ thuộc vào nhãn này.

Cho tới khi SELinux ra đời và được tích hợp vào Linux từ sau phiên bản 2.4, cơ chế kiểm soát truy nhập đã có những thay đổi. Thuật ngữ "SELinux" có thể được coi như tên gọi cho một cơ chế điều khiển truy nhập mới, nhưng nếu theo nghĩa thuần túy của nó thì SELinux đơn giản là " tăng cường bảo mật cho Linux". Như vậy, trong các tài liệu, SELinux được nhắc tới như một khái niệm mới về điều khiển truy nhập nhưng nó chính là "cơ chế MAC đã được gia cố để tăng thêm khả năng bảo mật".

Trong hệ thống sau khi kích hoạt SELinux, mỗi tài nguyên và chủ thể đều được gán cho một nhãn. Nhãn này là sự tổ hợp của 6 cơ chế điều khiển truy nhập: DAC, RBAC, TE, MLS, MCS và tất nhiên dựa trên cơ chế MAC. Điều đáng chú ý nhất ở SELinux là việc áp dụng cơ chế TE – Type Enforcement (thực thi dựa trên type/domain của đối tượng/chủ thể). Các nhà phát triển tùy theo mục đích xây dựng hệ thống sẽ tạo ra các role (vai trò), các type (kiểu) và gán cho mỗi User (người dùng). Sau đó công việc sẽ là viết các module policy nhằm ban quyền cho các chủ thể sở hữu những nhãn đó.

Nói đến đây, có một ưu điểm nổi bật của SELinux đó là khả năng điều khiển các chủ thể, mà các chủ thể có thể gộp chung lại chính là các process (tiến trình). Các process khi được tạo ra cũng được gán cho những Security context (các nhãn an ninh), trong các policy của SELinux sẽ có những dòng lệnh để quy định quyền và tiến trình làm việc của các process này. SELinux kiểm soát các chủ thể thông qua các process, và từ đây, SELinux có khả năng giải quyết những nguy cơ đề cập trong phần trước:

Thứ nhất, các doanh nghiệp sẽ không còn lo về vấn đề nhân viên sẽ gửi dữ liệu ra ngoài qua các trang web xã hội, hoặc các hòm thư điện tử nữa. Vì người quản trị chỉ cần chặn các trang web này từ chính máy tính của nhân viên thông qua một giải pháp rất đơn giản là cho những trang web này vào danh sách hạn chế của file hosts rồi bảo vệ file hosts bằng module policy không cho nhân viên truy cập và chỉnh sửa file này.

Thứ hai, đối với các ứng dụng như skype, yahoo chat… Nếu như trong doanh nghiệp hoặc công ty không tự xây dựng được một ứng dụng trao đổi nội bộ tương tự thì nhu cầu sử dụng skype và yahoo ngay cả khi đang làm việc cho mục đích công việc cũng là chính đáng, chưa nói tới việc không kiểm soát được các nhân viên khi đang ngoài giờ hành chính.Để khắc phục vấn đề này có hai giải pháp:

- Giải pháp thứ nhất chính là bảo vệ file dữ liệu bằng policy. Bản chất của việc gửi file thông qua các trang web hoặc các ứng dụng chính là một tiến trình có khả năng đọc (read), lấy thuộc tính (getattr) về dung lượng, tên… của file đó. SELinux kiểm soát nguy cơ này bằng cách kiểm soát các quyền giữa tiến trình đó và file mà tiến trình muốn thực hiện.
- Một giải pháp khác được sử dụng là tiện ích Sandbox của SELinux. Các ứng dụng này sẽ bị tắt các tính năng như gửi file đính kèm, chụp ảnh màn hình, quay camera hoặc chat voice. Khi chạy một ứng dụng với Sandbox, các process sẽ được đổi từ miền mặc định sang miền mới là `sandbox_t`, `sandbox_web_t`hoặc`sandbox_web_client_t`tùy vào từng phiên bản sandbox. Sau đó các domain này được kiểm soát bởi những policy mới, do đó ứng dụng Sandbox giúp kiểm soát các tính năng của mỗi ứng dụng bởi vì bản chất mỗi tính năng này đều tạo ra một process, process này sẽ được kiểm soát bởi các policy của SELinux, do đó SELinux có thể hoàn toàn cho phép hay ngăn chặn những gì mà doanh nghiệp mong muốn.

Thứ ba, đối với vấn đề kiểm soát truy cập. SELinux cung cấp một cơ chế kiểm soát truy cập rất chặt chẽ thông qua các policy. Mỗi thành viên trong doanh nghiệp tùy vào công việc, nhiệm vụ và vị trí sẽ được gán cho những security context, sau đó các module policy sẽ được xây dựng để kiểm soát truy cập đối với những context này. Bản chất của việc điều khiển truy cập chính là điều kiểm soát những quyền mà một Subject có đối với một Object. SELinux gắn cho Subject và Object những nhãn và sử dụng các policy để quy định những tương tác (action) giữa các nhãn này. Khi đó tức là một

process tác động lên một file hoặc tài nguyên đã được kiểm soát, vì vậy những nguy cơ về bảo mật hoàn toàn có thể được giải quyết.

Để việc thực hiện kịch bảo bảo mật cho doanh nghiệp rút gọn này được chặt chẽ, một số giả thiết sẽ được áp dụng:

1) TGĐ và các thành viên trong HĐQT mặc dù có quyền hạn cao nhưng đối với các tài liệu cần bảo mật cao cũng chỉ có quyền xem (open, read) và không được thực hiện bất cứ quyền hạn nào khác. Nếu muốn phải thông qua phòng ban phụ trách công việc tương ứng.
2) Mỗi phòng bàn có 1 folder lưu trữ tài liệu riêng, nhân viên của phòng ban nào chỉ vào được folder tương ứng, muốn vào folder khác phải được phê duyệt và cấp quyền nếu không sẽ bị coi là truy cập trái phép và bị từ chối truy cập.
3) Trong một phòng ban, chỉ có nhân viên phụ trách 1 tài liệu nhất định mới có toàn quyền (đọc, ghi, xóa) đối với tài liệu đó. Quyền xóa vẫn bị giới hạn đối với một số tài liệu nhất định. Các nhân viên khác và ngay cả trưởng phòng cũng chỉ có quyền đọc tài liệu đó.
4) Nếu một nhân viên trong 1 phòng, thậm chí là trưởng phòng có công việc cần phải sử dụng một quyền khác (ngoài đọc) đối với 1 tài liệu, thì phải được phê duyệt của trưởng phòng và người quản trị sẽ kích hoạt hệ thống cho phép nhân viên đó thực hiện. Nhưng khuyến khích thực hiện quyền này thông qua người phụ trách chính.
5) Nhân viên của một phòng ban nếu muốn đọc tài liệu của phòng ban khác cũng cần xin phê duyệt và cấp quyền.
6) Không cấp quyền copy cho bất cứ thành viên nào.
7) Đổi type đối với mỗi tài liệu để kiểm soát được chặt chẽ, tránh trường hợp nhân viên có thể gửi tài liệu ra ngoài (ví dụ đổi type sẽ chặn được nhân viên gửi tài liệu qua mail, facebook…)

8) Thành viên nào cố tình vi phạm sẽ đều được AVC log lại quá trình truy cập trái phép, tức là sẽ bị phát hiện. tùy vào việc giải trình, mức độ xử phạt do doanh nghiệp quy định.

Bảo mật trong doanh nghiệp suy cho cùng chính là gồm hai yêu cầu là bảo mật dữ liệu ở chung trên máy Server sau khi các Client đã truy cập vào và bảo mật dữ liệu riêng trên các máy Client. Cả hai yêu cầu này đều có thể thực hiện được bằng khả năng kiểm soát truy cập của SELinux mà cụ thể chính là 3 giải pháp vừa trình bày trên. Không mất tính tổng quát, dưới đây sẽ là những phân tích và thực hiện kiểm soát truy cập mà một người dùng bất kì, đại diện cho một thành viên trong doanh nghiệp, muốn truy cập tới một tài liệu cần bảo mật. Những kết quả kiểm soát truy cập tiếp theo chính là minh chứng cho tính khả thi của giải pháp mà đồ án đề xuất và phân tích.

3.2 Gán nhãn

Công việc gán nhãn có thể chia làm hai phần, đầu tiên là phân tích cơ cấu tổ chức của doanh nghiệp và xác định quyền hạn của mỗi thành viên đối với mỗi tài nguyên cần bảo mật. Tiếp theo là tiến hành gán nhãn trực tiếp trong hệ thống. Để tiện cho quá trình trình bày, từ đây, các thành viên trong doanh nghiệp sẽ được gọi chung là các chủ thể (Subject) và các file, tài nguyên sẽ được gọi là các đối tượng (Object).

Theo như sơ đồ cơ cấu tổ chức đơn giản của Công ty cổ phần T đã nêu ở trên thì cơ cấu này gồm:

- Hội đồng quản trị:
  - Chủ tịch HĐQT
  - Các cổ đông nằm trong HĐQT
- Tổng giám đốc.
  - Phó tổng giám đốc
  - Các trợ lý tổng giám đốc
- Các phòng ban

- Phòng kinh doanh
  - Trưởng phòng kinh doanh
  - Các nhân viên
- Phòng kế toán – tài chính:
  - Kế toán trưởng
  - Các kế toán viên
- Phòng kĩ thuật
  - Trưởng phòng kĩ thuật
  - Các kĩ thuật viên

Đứng từ góc độ các chủ thể có thể tác động lên đối tượng cần bảo mật, và giả sử:

- Đối tượng cần bảo mật là dữ liệu X, thuộc quyền kiểm soát của phòng 1 (tên phân biệt với phòng 2, phòng 3…).
- Các thành viên trong HĐQT và ban giám đốc có cùng vai trò, quyền hạn đối với đối tượng X.
- Trong các phòng ban khác nhau thì tài liệu thuộc công việc của phòng ban nào do phòng ban đó phụ trách, các phòng ban khác không có quyền hạn gì đối với dữ liệu đó.
- Trong một phòng ban, chỉ có 1 nhân viên toàn quyền phụ trách đối tượng X, còn các nhân viên khác không được toàn quyền.

thì cơ cấu tổ chức sẽ được rút gọn như sau để tiện cho quá trình phân quyền và gán nhãn như trong bảng 3.1.

Đến đây các chủ thể đã được gán cho các tài khoản để đăng nhập vào máy tính trong doanh nghiệp. Sau khi đăng nhập vào hệ thống, các chủ thể sẽ được gán các Security context (các nhãn) để SELinux kiểm soát thông qua các modulepolicy. Việc gán nhãn dựa trên các yếu tố sau:

- Gán SELinux user: tùy theo quyền hạn của chủ thể đối với hệ thống để gán SELinux user. Ví dụ nhân viên sẽ được gán staff_u, người quản trị hệ thống được gán sysadm_u…

```
root # semanage login -l

Login Name                SELinux User

__default__               user_u
root                      root
swift                     staff_u
system_u                  system_u
```

Hình 3. 1. Minh họa mối quan hệ giữa chủ thể với SELinux user.

- Gán role (vai trò): Dựa theo đặc thù về vị trí, công việc, trách nhiệm mà các thành viên sẽ được gán các role khác nhau. Một SELinux user có thể có một hoặc nhiều role. Ví dụ một nhân viên phòng kĩ thuật vừa có vai trò là nhân viên, vừa có vai trò là người quản trị. Do đó anh ta được gán 2 role là staff_r và system_r.

```
root # semanage user -l

SELinux User    SELinux Roles

root            staff_r sysadm_r
staff_u         staff_r sysadm_r
sysadm_u        sysadm_r
system_u        system_r
unconfined_u    unconfined_r
user_u          user_r
```

Hình 3. 2. Minh họa mối quan hệ giữa SELinux user và role

- Gán type/domain: type/domain xác định chi tiết công việc mà một chủ thể sẽ làm và được xác định dựa trên vai trò của chủ thể đó. Ví dụ một nhân viên kế toán sẽ làm công việc như thống kê các khoản thu, chi, và tính toán công nợ. Do đó một role sẽ ứng với một hoặc nhiều type khác nhau.

Bảng 3. 1. Cơ cấu rút gọn của doanh nghiệp nhìn từ góc độ đối tượng X

Stt	Chủ thể	Linux accounts	Giải thích
1	Thành viên Ban quản trị	*Ceo*	*Coi các thành viên trong HĐQT và TGĐ, PTGĐ, các trợ lý có quyền hạn ngang nhau đối với các dữ liệu cần bảo mật*
2	Trưởng phòng 1	*Manager1*	*Phòng 1 có toàn quyền phụ trách dữ liệu X*
3	Nhân viên phụ trách 1	*Staff1_major*	*Nhân viên của phòng 1 phụ trách công việc liên quan tới tài liệu cần bảo mật X*
4	Nhân viên không phụ trách 1	*Staff1_minor*	*Nhân viên phòng 1 nhưng không phụ trách công việc trên*
5	Trưởng phòng 2	*Manager2*	*Phòng 2 không có quyền hạn đối với dữ liệu X*
6	Nhân viên 2	*Staff2*	*Nhân viên của phòng 2*

SELinux cho phép ánh xạ từ một tài khoản chủ thể tới một SELinux user, một user được gán cho một hoặc nhiều role, một role lại được gán cho một hoặc nhiều type/domain. Và suy cho cùng thì chính là ánh xạ từ một chủ thể tới một hoặc nhiều type/domain. Đó chính là phép ánh xạ từ một chủ thể tới một Security context. Theo như cách các policy được xây dựng thì hai các type/domain sẽ xác định quyền hạn của một chủ thể. Theo phép ánh xạ này thì rõ ràng hai hoặc nhiều nhân viên có thể có cùng quyền hạn. Điều đó là dễ hiểu vì theo đặc thù công việc của họ, họ có chung vai trò, vị trí, trách nhiệm và nhiệm vụ.

Quay trở lại với doanh nghiệp đang được phân tích. Để gán nhãn cho các chủ thể, cần xác định trước tiên là quyền hạn của họ đối với một đối tượng. Vẫn với những giả sử bên trên, đối tượng được xét là dữ liệu X. Và giả sử rằng cần bảo vệ dữ liệu X từ 3 yêu cầu là Read (đọc), Write (cập nhật, sửa thông tin), Delete (xóa file dữ liệu). Bảng 3.2 sau đây là bảng phân quyền cho các chủ thể mà bây giờ được đại diện bởi các SELinux user khi xét đến khả năng tác động lên đối tượng X.

Bảng 3. 2. Phân quyền cho các chủ thể đối với đối tượng dữ liệu X

Stt	SELinux user	Read	Write	Delete
1	Ceo	Y	N	N
2	Manager1	Y	N	N
3	Staff1_major	Y	Y	Y
4	Staff1_minor	Y	N	N
5	Manager2	N	N	N
6	Staff2	N	N	N

Hai chủ thể có cùng quyền hạn trong bảng trên chưa chắc đã có cùng nhãn, vì bảng trên xét đối với đối tượng X , còn trong trường hợp đối tượng là Y thì sẽ có sự khác nhau. Ví dụ Ceo và Manager1 cùng có tập quyền hạn đối với X, nhưng đối với dữ liệu Y thuộc phòng 2 phụ trách thì lúc này Manager1 lại không có quyền hạn gì đối với Y. Chính vì thế, có thể coi policy là phép tổ hợp của đối tượng và chủ thể để đưa ra những "điều luật" hợp lý cho quá trình kiểm soát truy nhập.

Công đoạn phân tích tổ chức đến đây kết thúc, người quản trị đã biết được thành viên nào có vai trò gì, làm công việc gì và có quyền hạn gì. Tiếp theo sẽ gán nhãn cho các chủ thể. Các nhãn này có thể được định nghĩa, tên các nhãn thường mang tính chất

gợi ý, thể hiện tính chất công việc của chủ thể. Bảng sẽ trình bày quá trình gán nhãn cho các chủ thể đã phân tích bên trên:

Bảng 3. 3. Gán nhãn cho các chủ thể.

Stt	SELinux user	User (_u)	Role(_r)	Type/domain(_t)
1	Ceo	ceo_u	ceo_r	ceo_t
2	Manager1	manager_u	manager_r	manager1_t
3	Staff_major1	staff_u	staff1_r	staff1_major_t
4	Staff_minor1	staff_u	staff1_r	staff1_minor_t
5	Manager2	manager_u	manager_r	manager2_t
6	Staff2	staff_u	staff2_r	staff2_t

Từ đây chúng ta thấy được ưu điểm nổi bật trong kiểm soát các chủ thể của SELinux so với cơ chế DAC thông thường. Đó chính là sự linh hoạt và nhanh gọn nhưng vẫn đảm bảo chặt chẽ trong bảo mật. Những chủ thể có cùng quyền hạn sẽ được gán chung một type/domain. Và quá trình quản lí sẽ diễn ra hiệu quả hơn là quản lý tới tùng User một. Mà thành phần quan trọng nhất là type để áp dụng các lệnh allow trong khi tạo policy. Quá trình tạo policy sẽ được trình bày ở phần tiếp theo của chương.

3.3 Tạo policy

3.3.1 Một số cú pháp quan trọng trong ngôn ngữ policy

Để "hiện thực hóa" giải pháp cho bài toán "***nâng cao bảo mật dữ liệu trong doanh nghiệp***", trong phần này, các chính sách về bảo mật sẽ được xây dựng để hệ thống máy tính có thể hiểu và quản lý được các tiến trình truy cập. Các chính sách đó được số hóa thông qua các Security policy mà được tạo ra và biên dịch rồi chèn vào nhân của hệ thống. Đầu tiên là một số cú pháp đơn giản trong ngôn ngữ policy (Policy language).

❖ **Đặt tên cho module policy:**

Đầu tiên cần khai báo tên cho policy với cú pháp:

```
policy_module(policy_name, x.x.x);
```

Trong đó:

- `policy_module`: từ khóa khai báo tên một policy mới
- `policy_name`: tên muốn đặt cho policy này
- `x.x.x`: phiên bản của policy

❖ **Khai báo một type/domain mới:**

Cú pháp:

```
type newtype_t;
```

Câu lệnh bên trên cho phép khai báo một tên miền hoặc tên kiểu mới cho một đối tượng hoặc một chủ thể. Trong một module policy, tất cả những type/domain được sử dụng đều phải được khai báo (nếu như đó là type/domain mới và chưa có trong nhân), hoặc được gọi ra với cú pháp:

```
require { type type1_t, type2_t, type3_t; }
```

❖ **allow rule :**

Cú pháp:

```
allow Source_type Target_type : Class { Permission };
```

Bảng 3.4 dưới đây giải thích các từ khóa được dùng trong câu lệnh trên.

Quy tắc này định nghĩa các permission mà proccess có nhãn source_type được phép thi hành trên các class có nhãn Target_type. "allow" rule là câu lệnh phổ biến nhất trong hầu hết các module policy.

Ví dụ: `allow unconfined_t  mytype_t:file{ read getattr};`

Câu lệnh trên cho phép một chủ thể có nhãn unconfined_t tiến hành đọc và lấy những thuộc tính như ngày, thời gian… của một file có nhãn mytype_t.

Bảng 3. 4. Giải thích các từ khóa ngôn ngữ policy

`allow`	Từ khóa allow, quy định cấp một hoặc nhiều Permission cho một chủ thể khi tương tác với một đối tượng.
`Source_type`	Domain của chủ thể đang tiến hành truy cập tới một đối tượng.
`Target_type`	Kiểu của đối tượng đang được truy cập tới.
`Class`	Lớp của đối tượng (có thể là file, pipe, socket...).
`Permission`	Các quyền mà tiến trình có mieend là Source_type có thể thực thi trên các đối tượng có kiểu Target_type.

❖ **Câu lệnh đổi kiểu cho đối tượng :**

Cú Pháp:

```
type_transition Source_type Target_type : Class { new_type };
```

Bảng 3. 5. Giải thích các từ khóa ngôn ngữ policy

`type_transition`	Từ khóa type_transition cho câu lệnh đổi kiểu
`Source_type`	Miền của chủ thể
`Target_type`	Kiểu của đối tượng
`Class`	lớp của đối tượng (có thể là file, pipe, socket...)
`new_type`	type mới được thay thế cho đối tương

Quy tắc này định nghĩa khi một process thuộc Source_type tạo một file trong một thư mục có nhãn là Target_type, theo mặc định file được tạo ra phải có nhãn Target_type của thư mục mẹ chứa nó, nhưng nó lại được đặt nhãn new_type thay thế.

Ví dụ: `type_transition sshd_t tmp_t:file sshd_tmp_t;`

Câu lệnh trên còn tương đương với câu lệnh :

```
file_type_auto_trans(sshd_t, tmp_t, sshd_tmp_t);
```

❖ **Domain transitions for process :**

Cú Pháp :

```
type_transition Source_type Target_type : process new_type;
```

Quy tắc này định nghĩa khi một process có nhãn Source_type thực thi một file có nhãn Target_type, thì nhãn của process sẽ được chuyển thành new_type. Cú pháp này chỉ khác câu lệnh Type transitions for object ở chỗ thay class thành process.

Ví dụ: `type_transition sshd_t shell_exec_t:process user_t;`

Câu lệnh trên còn tương đương với câu lệnh :

```
domain_auto_trans(sshd_t, shell_exec_t, user_t);
```

❖ **Khai báo một vai trò (role) mới**

Câu lệnh Role có thể dùng để khai báo một định danh role hoặc có thể dùng để liên kết một định danh role với một hoặc nhiều types (có nghĩa là cho quyền role truy cập vào một hay nhiều miền (domain)).

Cú Pháp:

```
role role_id; //khai báo.
role role_id types type_id; // liên kết ROLE với TYPE.
```

Bảng 3. 6. Giải thích các từ khóa ngôn ngữ policy

`role`	Từ khóa role khai báo định danh vai trò
`role_id`	Định danh role được khai báo.
`types`	từ khóa không bắt buộc types
`type_id`	Khi sử dụng từ khóa types, một hoặc nhiều type_id có thể liên kết với một role_id. trong trường hợp có nhiều type_id ta đặt chúng trong dấu {}.

Ví dụ:
```
role myrole_r;
role myrole_r types mytype_t ;
```

3.3.2 Xây dựng các module policy

Như đã trình bày trong những phần trước, khi hệ điều hành đã tích hợp thêm SELinux thì tất cả quá trình truy cập đến các file, các tài nguyên của hệ thống đều được kiểm soát bởi cơ chế SELinux này. SELinux kết hợp hai vòng kiểm soát là cơ chế DAC và MAC đã được tăng cường bảo mật. Quá trình kiểm soát truy cập sẽ đưa ra quyết định cho phép hoặc từ chối dựa trên các module policy. Vì vậy, việc xây dựng các module policy này là rất quan trọng. Trong phần này, các module sẽ lần lượt được xây dựng để thực hiện các giải pháp đã đưa ra.

- **Kiểm soát truy cập**:

Để chứng minh cho tính khả thi của giải pháp bảo mật bằng SELinux nói chung và điều khiển truy cập trong mô hình Client-Server nói riêng, hai chủ thể tương ứng với 2 thành viên trong mô hình doanh nghiệp đã giới thiệu sẽ được tạo ra và được kiểm

soát thông qua policy, đó là Trưởng phòng 1 (manager1) và nhân viên phòng 1 chuyên phụ trách làm việc với tài liệu X này (staff1_major).

file manager1.te

```
# define a new user (manager1) and his context:
policy_module(manager1, 1.0.0)
#define manager1 role
role manager1_r;
# just allow read and open:
allow manager1_t staff1_major_t : file { read open };
# generate manager1_u
userdom_unpriv_user_template(manager1)
gen_user(manager1_u, user, manager1_r, s0, s0)
#EOF
```

file staff1_mojor.te

```
# define a new user (staff1_major) and his context:

policy_module(staff1_major, 1.0.1)

# required type:
require {
  type
       staff1_major_t,
       home_root_t,
       fs_t,
       unconfined_t;
  };
# define staff1_major role and type
role staff1_major_r;

# allow rules for read, write....:
```

```
allow staff1_major_t home_root_t : file { read open
setattr write unlink create };
allow staff1_major_t staff1_major_t : file { read open
setattr write unlink};

allow staff1_major_t home_root_t : dir { write add_name
remove_name};

# allow relabel to staff1_major_t
allow unconfined_t staff1_major_t : file { relabelto };
allow staff1_major_t fs_t : filesystem { associate };


# generate user staff1_major_u
userdom_unpriv_user_template(staff1_major)
gen_user(staff1_major_u, user, staff1_major_r, s0, s0)

#EOF
```

- **Kiểm soát các ứng dụng bằng Sandbox**

Sandbox cho phép hạn chế tối thiểu quyền hạn của các ứng dụng. Khi đó, một ứng dụng muốn thực hiện truy cập nào phải được định nghĩa bởi policy. Dưới đây là một ví dụ về module sandbox_gedit.te. Module này cho phép tiện ích gedit trong Unix/Linux có thể mở, đọc, ghi, lấy các thuộc tính… từ một file có nhãn `home_root_t, unconfined_t` hoặc`usr_t`.

file sandbox_gedit.te

```
# define a new user (staff1_major) and his context:
policy_module(sandbox_gedit, 1.0.0)
```

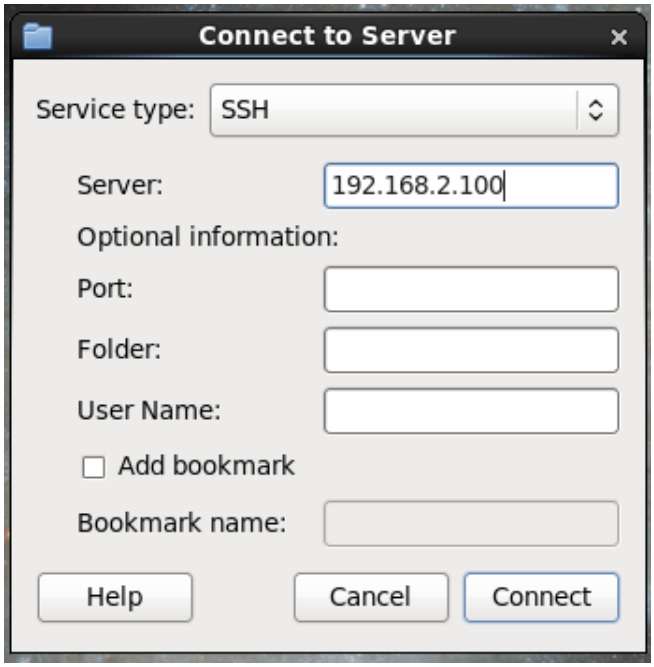
```
require {
            type sandbox_web_client_t,
            sandbox_web_t,
            user_devpts_t,
            home_root_t,
            usr_t,
            unconfined_t;
            };

#allow rules:
allow sandbox_web_client_t home_root_t : file { read
open setattr write execute};

allow sandbox_web_client_t unconfined_t : file { read
open setattr write execute};

allow sandbox_web_client_t usr_t : file { read open
setattr write execute execute_no_trans execmod};
allow sandbox_web_t user_devpts_t : chr_file { open };
#EOF
```

Sau khi được biên dịch và chèn vào hệ thống, các module policy này sẽ có trong danh sách các policy của SELinux.

3.4 Kiểm soát truy cập với SELinux

Sau khi các policy đã được cập nhật và chèn vào hệ thống, quá trình kiểm tra các phiên truy cập của hai chủ thể là manager1 và staff1_major được tiến hành. Dưới đây là các hình ảnh minh họa cho kết quả kiểm soát truy cập được tiến hành bởi hệ thống SELinux:

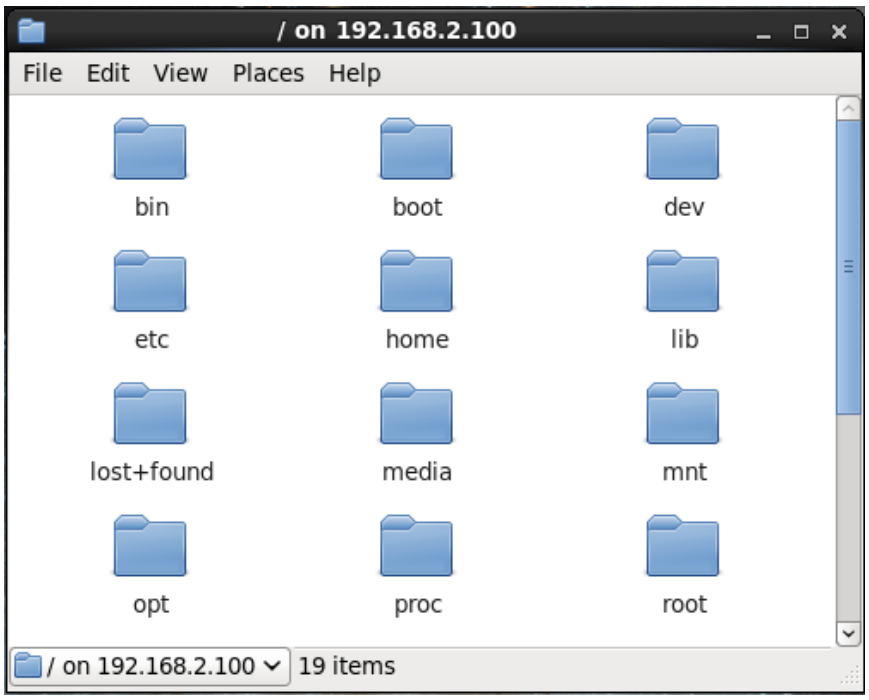
- Đầu tiên máy Client sẽ tiến hành kết nối tới máy Server có địa chỉ IP là 192.168.2.100 như trong hình 3.3. Giao thức ở đây được sử dụng là SSH (Secure Shell), đây là giao thức thực hiện phiên kết nối bảo mật giữa hai hệ điều hành Linux:



Hình 3. 3. Bắt đầu phiên kết nối tới máy Server để truy cập tài liệu

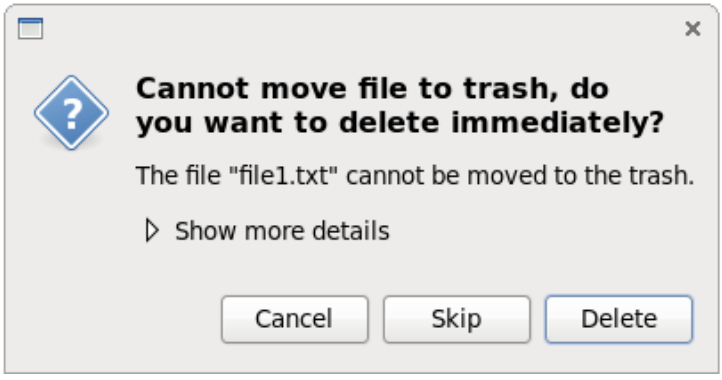
Khi nhấn nút "connect" thì hệ thống sẽ yêu cầu nhập vào User account và password để xác thực phiên truy cập. Một lưu ý là khi một Client truy cập vào một Server thì Client đó được coi như một User trong máy Server đó. Do đó trong máy Server cũng cần tạo một User mới tương ứng với máy Client (có cùng tên và mật khẩu), đồng thơi có hệ thống policy giống với máy Client để kiểm soát truy cập cho Client tương ứng.

- Kết quả kết nối thành công tới Server 192.168.2.100. Server là "kho dữ liệu" chứa các tài liệu mà các Client sẽ truy cập tới để thực hiện công việc. Mình họa quá trình truy cập thành công ở hình 3.4

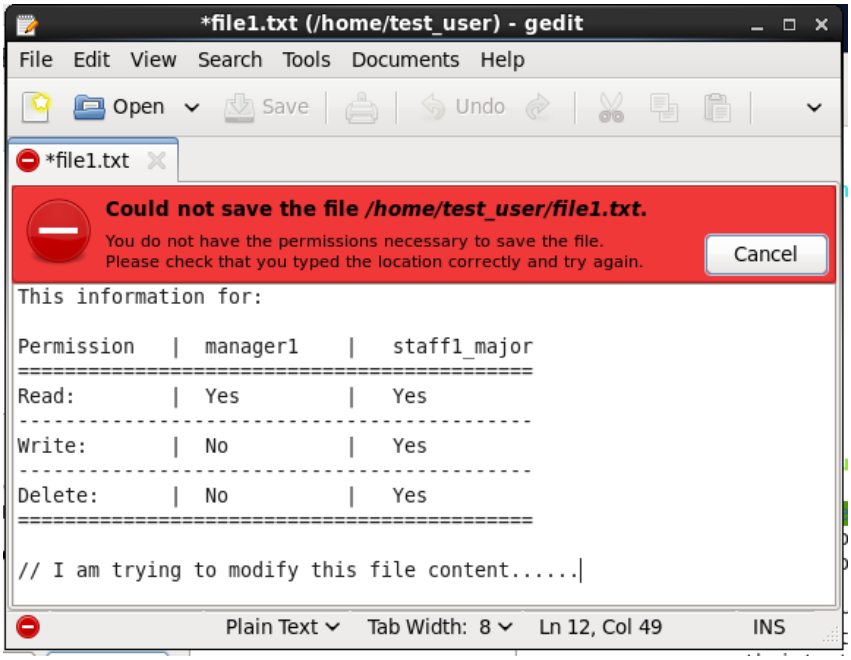


Hình 3. 4. Máy Client truy cập thành công tới máy Server

- Chủ thể manager1 bắt đầu thực hiện truy cập tới các dữ liệu mà đại diện ở đây là file1.txt. Các hình3.5,3.6 minh họa kết quả kiểm soát truy cập của SELinux đến máy thành viên NguyenVanA (manager1):

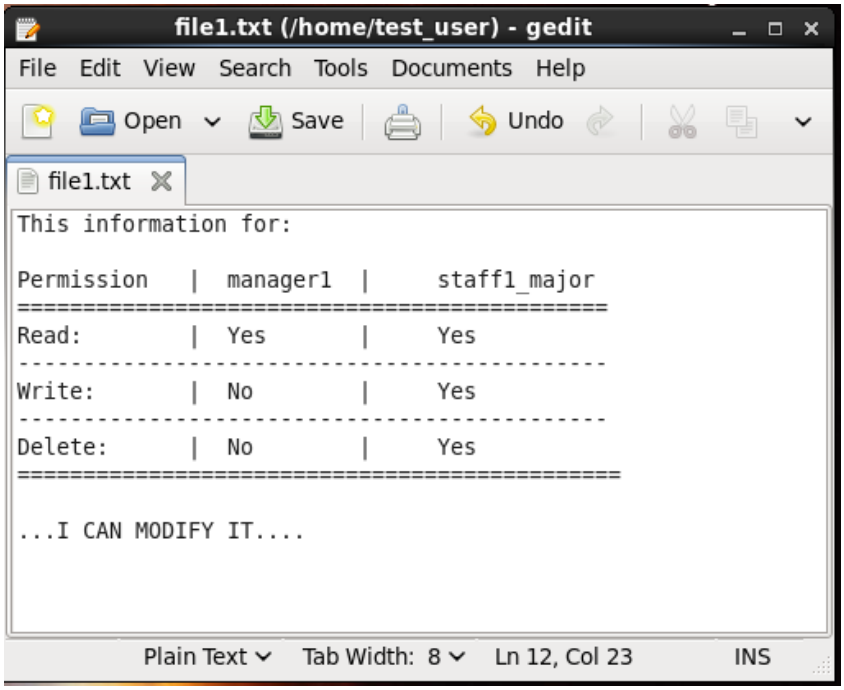


Hình 3. 5. Chủ thể manager1 không có quyền xóa (delete)



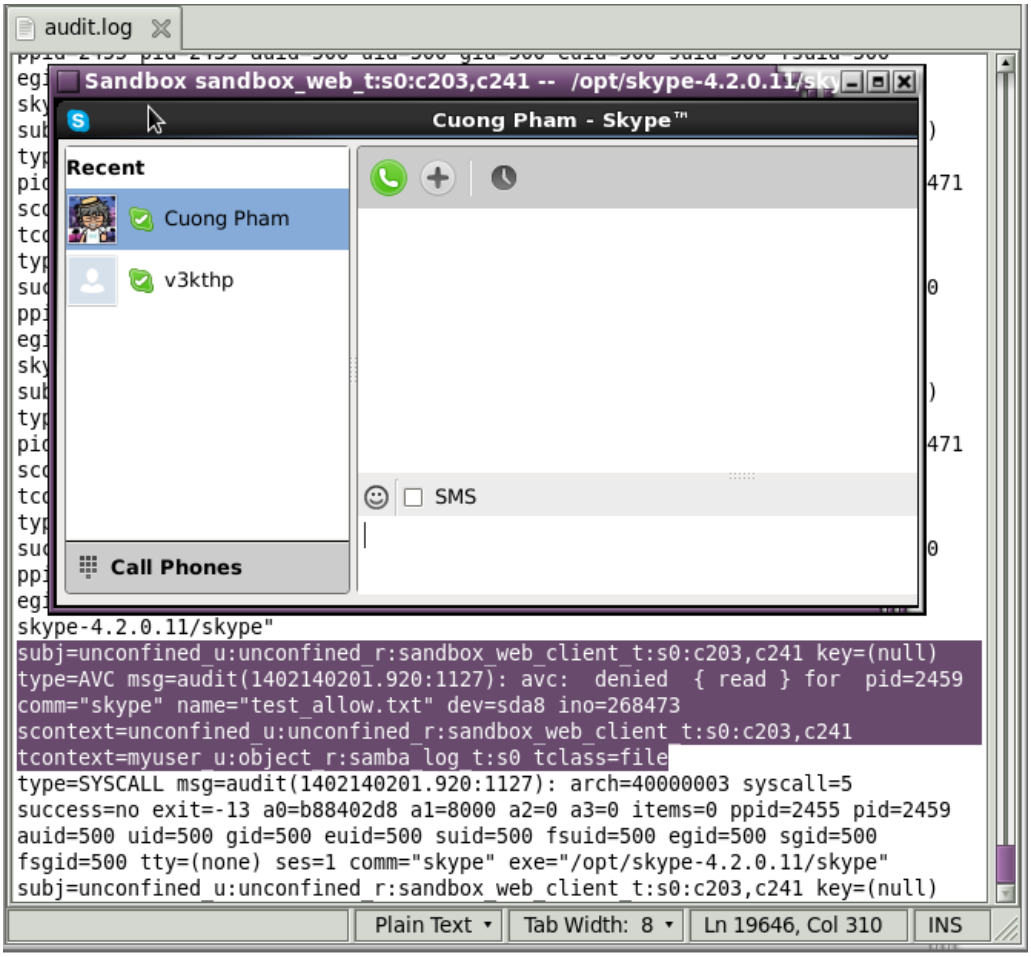
Hình 3. 6. Chủ thể manager1 không có quyền ghi (write)

- Hình 3.7minh họa sự kiểm soát truy cập của SELinux đối với máy của thành viên NguyenThiHa (staff1_major). Chủ thể staff1_major có quyền đọc, ghi, xóa đối với tài liệu X (file1.txt):

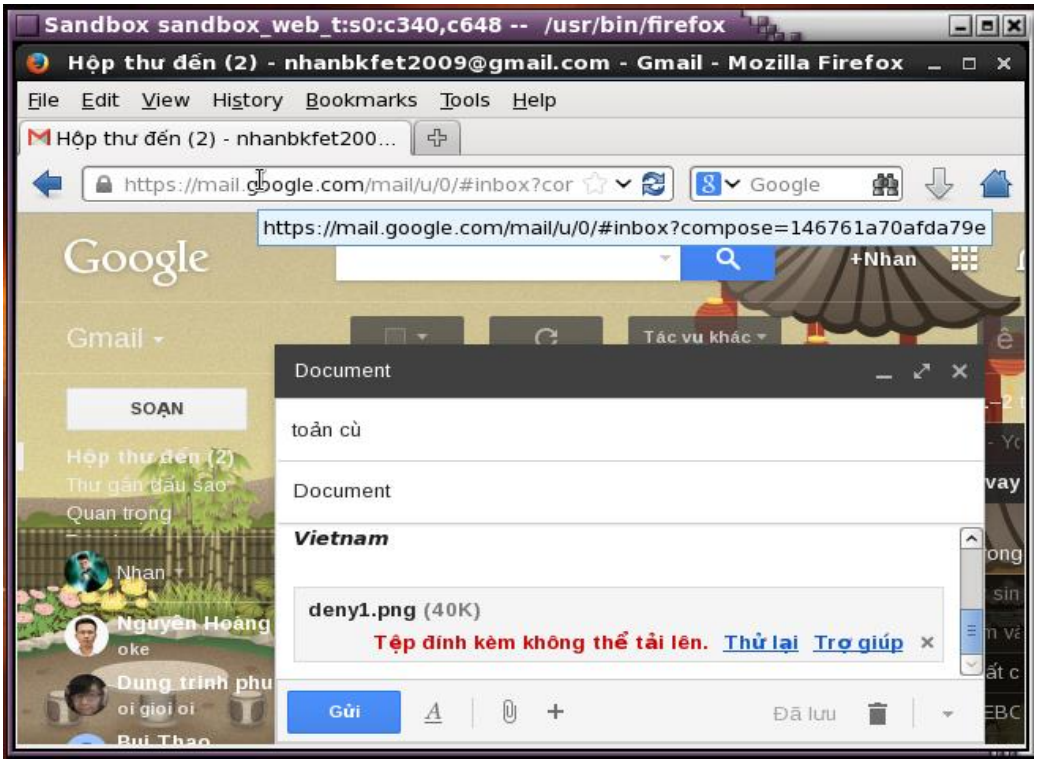


Hình 3. 7. Chủ thể staff1_major có quyền ghi (write) vào file1.txt

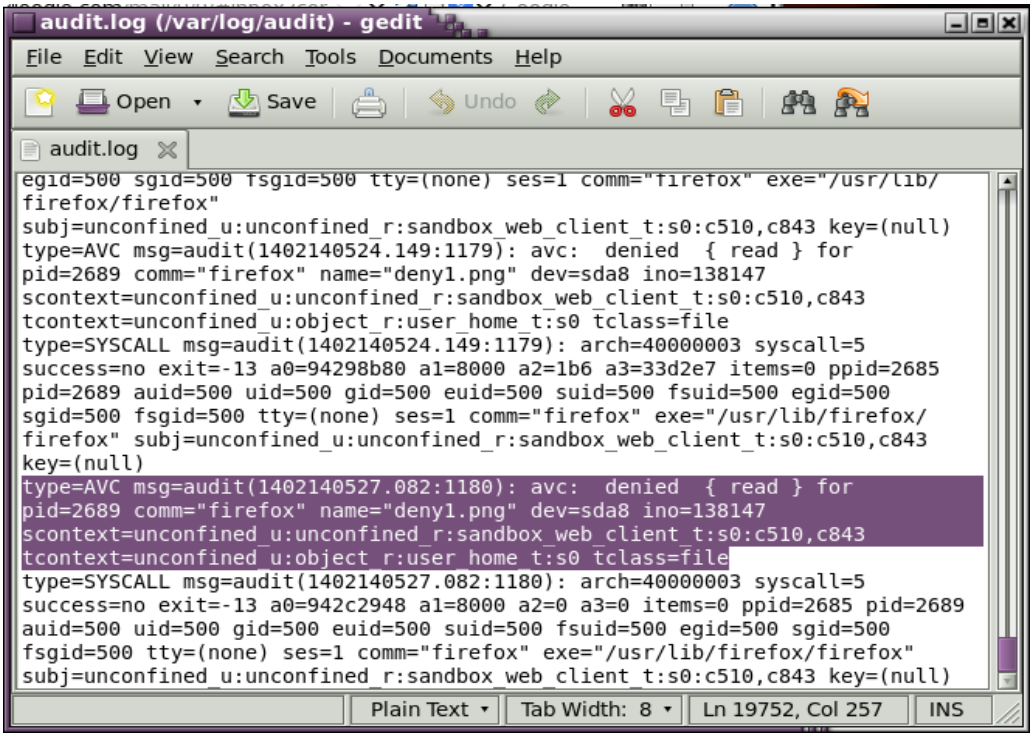
- Sandbox là giải pháp mới của SELinux cho phép chạy các ứng dụng dưới kiềm kiểm soát chặt chẽ của SELinux, mỗi tiến trình được sinh ra bởi các ứng dụng được chuyển sang domain mới là sandbox_web_client_t và được kiểm soát chặt chẽ với những policy riêng dành cho Sandbox. Tuy nhiên có một hạn chế ở đây là Sandbox mới được phát triển với mục đích người dùng tránh được những truy cập trái phép chứ chưa được tích hợp thẳng vào nhân để bắt buộc các nhân viên trong doanh nghiệp phải sử dụng. Dưới đây là hình ảnh minh họa kết quả kiểm soát ứng dụng skype và trình duyệt firefox bằng sandbox:



Hình 3. 8. Người dùng không gửi được file khi sử dụng Skype trong sandbox



Hình 3. 9. Người dùng không gửi được file khi sử dụng Gmail trong sandbox



Hình 3.10. Truy cập trái phép bởi Firefox được AVC log ghi lại

3.5 Kết luận

Trong bảo mật tài liệu, các doanh nghiệp phải đối mặt với một số vấn đề chính là truy cập trái phép và rò rỉ thông tin. Quá trình khắc phục các nguy cơ này được thực hiện thông qua những lợi thế của SELinux. Đầu tiên, cơ cấu tổ chức của doanh nghiệp sẽ được phân tích, rồi dựa trên quyền hạn, công việc, mỗi thành viên trong doanh nghiệp sẽ được gán cho một nhãn (security context) đại diện cho quyền hạn của mình. Thông qua các nhãn này, các policy quy định quyền hạn cho các thành viên đối với các tài liệu cần bảo mật. Quá trình truy cập chỉ được phép khi đã được quy định trong các policy, do đó yếu tố kiểm soát truy cập là hoàn toàn chặt chẽ. Thêm vào đó, ứng dụng sandbox của SELinux cũng khắc phục hiệu quả nguy cơ rò rỉ thông tin. Các nhân viên trong công ty, doanh nghiệp nên được khuyến khích sử dụng ứng dụng này đối với một số trình duyệt và phần mềm có thể đe dọa đến các tài liệu cần bảo mật.

KẾT LUẬN

SELinux, đúng như tên gọi của nó, là sự tăng cường bảo mật cho hệ điều hành Linux và vẫn được biết đến và sử dụng như giải pháp bảo mật cho máy tính cá nhân. Trong đề tài này, SELinux đã được mạnh dạn giới thiệu và ứng dụng trong doanh nghiệp. Mặc dù đã rất cố gắng, nhưng do những hạn chế trong khả năng nghiên cứu, kinh nghiệm làm việc và những kiến thức về lĩnh vực bảo mật, nên chúng tôi không thể tránh được những thiếu sót nhất định. Trong đó, các vấn đề ở mức độ giải pháp đã được giải quyết và được chứng minh tính khả thi, tuy nhiên một hệ thống hoàn chỉnh để áp dụng vào một doanh nghiệp cụ thể vẫn chưa xây dựng được, từ đó phân tích và đánh giá hiệu năng cũng như những ưu nhược điểm của giải pháp này khi làm việc thực tế.

Cho đến nay SELinux vẫn còn khá xa lạ và khó sử dụng đối với nhiều người. Tiếp nối đồ án này, chúng tôi hi vọng sẽ có được sự giúp đỡ của nhiều nhà phát triển và các kĩ sư yêu thích lĩnh vực này để có thể xây dựng được một bộ công cụ quản lý bảo mật dữ liệu trong doanh nghiệp. Người dùng sẽ không còn phải thêm user, role, domain/type bằng các dòng lệnh trong Terminal nữa, cũng sẽ không còn phải viết policy bằng việc tạo từng file một nữa, chúng tôi mong muốn sẽ xây dựng lên một bộ công cụ trực quan, thân thiện và linh hoạt cho nhiều doanh nghiệp khác nhau đều có thể sử dụng một cách hiệu quả.

Là những sinh viên ĐHBK HN, chúng tôi luôn ý thức được vai trò và trách nhiệm của mình đối với công việc học tập và nghiên cứu ngay từ khi còn ngồi trên ghế nhà trường. Mai này, khi bước ra khỏi cánh cổng ĐHBK HN, trở thành những kĩ sư thực thụ, chúng tôi tự nhận thức được rằng kiến thức và kĩ năng của mình không bao giờ là hoàn hảo, do đó, trong cả môi trường làm việc và nghiên cứu chúng tôi vẫn sẽ cố gắng học hỏi và hoàn thiện bản thân để xứng đáng là một kĩ sư đã được học tập và rèn luyện trong một ngôi trường hàng đầu về kĩ thuật – trường Đại học Bách khoa Hà Nội.

TÀI LIỆU THAM KHẢO

[1] http://www.noip.gov.vn/ , truy cập cuối cùng ngày 17/5/2014.

[2] http://luatminhkhue.vn/ , truy cập cuối cùng ngày 17/5/2014.

[3] " Sáng chế, bằng độc quyền và thông tin sáng chế" , Lutz Mailander – Trưởng bộ phận Thông tin sáng chế, Ban Cơ sở hạn tầng sở hứu trí tuệ toàn cầu.

[4] http://www.dankinhte.vn/, truy cập cuối cùng ngày 19/5/2014.

[5] "Bí mật kinh doanh và phương thức bảo vệ", Luật sư Nguyễn Thanh Hà – Giám đốc công ty luật S&B

[6] http://www.thongtincongnghe.com/, truy nhập cuối cùng ngày 17/5/2014

[7] http://office.microsoft.com/, truy nhập cuối cùng ngày 17/5/2014

[8] http://luanvan.co/. truy nhập cuối cùng ngày 17/5/2014

[9] http://vi.wikipedia.org/ , truy nhập cuối cùng ngày 18/5/2014

[10] https://access.redhat.com/, truy nhập cuối cùng ngày 18/5/2014

[11] http://en.wikipedia.org/wiki/Security-Enhanced_Linux, truy cập cuối cùng ngày 15/4/2014

[12] The SeLinux Notebook - The foundations (trang 25)

[13] The SeLinux Notebook - The foundations (trang 25)

[14] The SeLinux Notebook - The foundations (trang 26)

[15] Role-Based Access Control Modelsyz Ravi S. Sandhu , Edward J. Coynek, Hal L. Feinsteink and Charles E. Youmank Revised October 26, 1995

[16]http://www.centos.org/docs/5/html/Deployment_Guide-en-US/sec-mls-ov.html, truy nhập lần cuối cùng ngày 15/4/2014

[17]http://www.centos.org/docs/5/html/Deployment_Guide-en-US/selg-overview.html, truy cập lần cuối cùng ngày 15/4/2014

[18] http://en.wikipedia.org/wiki/Multi_categories_security

[19] http://selinux-mac.blogspot.com/2009/06/multi-category-security.html

[20] Linux Security Modules: General Security Hooks for Linux ,Stephen Smalley NAI Labs

[21] https://www.ibm.com/developerworks/library/l-selinux/

[22] https://www.imperialviolet.org/2009/07/14/selinux.html, truy cập lần cuối cùng ngày 16/4/2014

[23] http://selinux-mac.blogspot.com/2009/09/avc-denials-example.html

[24]https://wiki.gentoo.org/wiki/SELinux/Tutorials/Where_to_find_SELinux_permission_denial_details, truy cập lần cuối cùng ngày 16/4/2014

[25] The SeLinux Notebook - The foundations(trang 65)..

[26] Báo cáo bảo mật thông tin (SECURITY ENHANCED LINUX)

[27] The SeLinux Notebook - The foundations (trang 16).