

기능명세서

(ADV_FSP.5)

- 고등급 보안 마이크로커널 개발 -

<목 차>

1. 소개	1
1.1. 문서의 목적	1
2. 기능 명세	3
2.1. FAU 서브시스템 내 TSFI	3
2.2. FRU 서브시스템 내 TSFI	5
2.3. FPT 서브시스템 내 TSFI	9
2.4. FIA 서브시스템 내 TSFI	11
2.5. FMT 서브시스템 내 TSFI	15
2.6. FMT 서브시스템 내 TSFI	17

1. 소개

본 문서는 TOE에 해당하는 CHAOS(Chibi-based High Assurance Operating System)의 TSFI(TOE Security Function Interface)를 서술한다. TSFI는 외부실체가 TSF(TOE Security Function)에게 데이터를 전송하거나 수신함으로써 TSF의 서비스를 호출할 수 있는 모든 수단을 의미한다. 본 문서를 활용하여 평가자는 TSF가 어떻게 SFR(Security Function Requirements)을 만족하는지 평가할 수 있다.

1장은 본 문서의 개요를 설명하며 2장에서는 본 문서에서 다루는 TOE의 각 기능이 동작하기 위해 필요한 기능을 인터페이스 측면에서 상세하게 서술한다.

1.1. 문서 목적

기능명세서는 외부 입력으로 인해 트리거되는 TSFI를 식별하고 이를 바탕으로 TOE가 어떻게 동작하는지 여부를 점검하기 위해 작성한다. 이를 통해 개발자와 평가자는 본 문서를 활용하여 향후 시험(테스팅)을 수행하는데 필요한 정보를 확인 할 수 있다. 왜냐하면 TSF와 상호작용하는데 활용되는 TSFI에 대한 적절한 정보를 제공받지 않고, TOE에 대한 적절한 시험을 수행하는 것은 제한되기 때문이다. 따라서 본 문서에서는 TSFI를 명세할 때, 아래 활용 목적, 호출 조건, 매개변수, 매개변수 설명, 인터페이스 행동, 오류 메시지 설명을 포함함으로써 평가자가 시험서에 대한 평가를 용이하게 수행할 수 있도록 작성하였다.









2. Unified Modeling Language (UML)

본 장에서는 TOE의 내부 구조를 표현하기 위해 사용되는 표기법인 UML에 대해 설명한다. UML은 국제 표준화 기구 중 하나인 Object Management Group(이하 OMG)에서 제정한 통합 모델링 언어이다. 해당 언어는 활용 목적에 따라 12개의 다이어그램을 작성할 수 있다. 이때 본 연구진은 외부 입력 별로 트리거되는 TSFI와 그로 인한 TOE의 동작을 준정형한 방식으로 표현하기 위해 준정형 기법 중 하나인 UML을 활용한다. UML은 활용 목적에 따라 12개의 모델(다이어그램)을 도출할 수 있다. 본 문서에서는 외부 입력으로 인해 트리거되는 TSFI를 식별하고 이를 바탕으로 TOE가 어떻게 동작하는지 여부를 의 행동을 준정형한 방법으로 표현하기 위해 상태기계 다이어그램을 활용한다. 상태기계 다이어그램은 외부 입력에 따른 시스템의 내부 상태 변화를 모델링하기 때문에, 외부입력에 따라 트리거되는 TSFI에 의해 동작하는 TOE의 동작을 표현하는데 적합하다.

다이어그램 명	활용 목적
클래스 다이어그램	시스템의 구조를 클래스 관점에서 모델링
오브젝트 다이어그램	특정 시점에서의 시스템 동작과 클래스 간 관계를 모델링
컴포넌트 다이어그램	시스템의 기능을 컴포넌트(1개 이상의 클래스 포함) 관점에서 모델링
복합 구조 다이어그램	컴포넌트 간 계층 관계와 상호작용을 모델링
유즈케이스 다이어그램	사용자가 활용할 수 있는 시스템의 기능을 모델링
액티비티 다이어그램	시스템 실행 흐름을 모델링
시퀀스 다이어그램	객체 간 주고받는 메시지 교환을 시간의 흐름에 따라 모델링
배포 다이어그램	시스템이 실제 하드웨어에 탑재된 후, 실행되는 흐름을 모델링
패키지 다이어그램	UML 내 다양한 요소를 목적에 따라 그룹화하여 모델링
타이밍 다이어그램	특정 시간에서의 개체 동작이나 오브젝트 상태를 모델링
상태기계 다이어그램	입력 값이나 외부 이벤트에 따라 변경될 수 있는 시스템 상태를 모델링
통신 다이어그램	시스템이나 객체 간 상호작용을 모델링

[표 1] UML 다이어그램

3장에 작성된 TOE의 상태기계 다이어그램의 구성 요소는 아래 표와 같다.

구성 요소 아이콘(명)	상세 설명
 (일반 상태)	실행 시, 제약 조건이 포함된 시스템 내부 상태
 (초기 상태)	시스템 초기 상태
 (조건)	상태 전이 시, 고려해야 하는 조건
 (상태 합병)	복수 상태들이 하나의 단일 상태로 병합
 (상태 분기)	하나의 상태에서 여러 가지 동시 상태로 분기
 (종료 상태)	외부 입력에 의해 시스템이 도달할 수 있는 최종상태
 (상태 전이)	외부 입력에 의해 발생하는 시스템 전이
 (상태 기계)	현재 상태 기계 내 상태 전이에 의해 발생할 수 있는 타 상태들의 집합

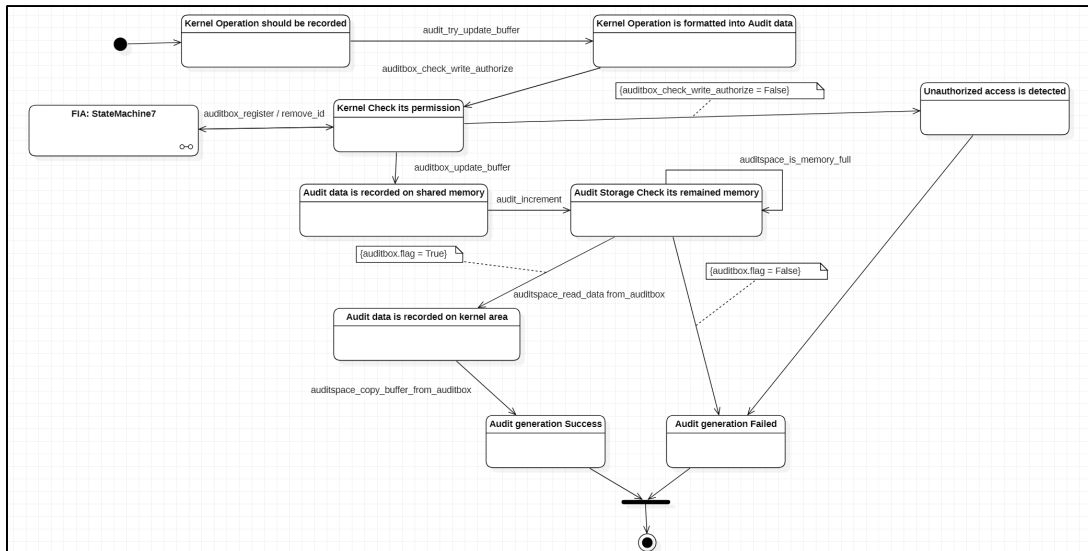
[표 2] 상태전이 다이어그램 구성 요소

3. 기능 명세

본 장에서는 TOE의 TSFI를 명세한다. TSFI는 사용자의 요청에 따라 TOE가 제공하는 서비스를 제공하기 위한 인터페이스인 시스템 콜 형태로 구현 및 사용된다.

3.1. FAU(감사) 관련 TSFI

본 절에서는 감사 관련 TSFI에 대해 서술한다. 감사 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.



[그림 9] 감사 모듈에 의한 TOE 세부 상태전이도

TOE는 사용자 요청을 처리하면서 발생하는 모든 동작에 대해 공유 메모리를 거쳐 감사 증적소에 저장한다. 해당 동작은 FAU 서브시스템에 수행된다. 사용자는 TOE에게 요청한 작업 내역을 audit_try_update_buffer 모듈을 통해 감사 증적소에 저장할 수 있는 포맷으로 변경한다. 이후 TOE는 auditbox_check_write_authorize 모듈을 통해 해당 사용자가 인가된 사용자인지 확인한다. 이때 인가될 수 있는 사용자 목록은 FIA 서브시스템 모듈 중 auditbox_register_id, auditbox_remove_id로 갱신된다. 인가된 사용자인 경우, TOE는 포맷에 맞게 변경된 작업 내역을 auditbox_update_buffer를 통해 공유메모리에 저장한다. 이후 TOE는 현재 감사 증적소의 잔여 메모리 양을 자체 점검한다. 이를 위해 audit_increment 모듈과 auditspace_is_memory_full 모듈이 트리거된다. TOE는 audit_increment 모듈에 의해 감사기록이 증적될 경우의 잔여 메모리 양을 auditspace_is_memory_full로 점검한다. 추가적으로 TOE는 auditspace_read_data_from_auditbox 모듈을 통해 auditbox.flag를 점검한다. 해당 값이 True 인 경우, TOE는 auditspace_copy_buffer_from_auditbox 모듈을 통해 공유 메모리에 저장된 데이터를 커널 내 감사 증적소에 저장한다.

자원 활용 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	audit_try_update_buffer	*function_name: 실행된 함수 명 is_success: 실행 성공 여부	감사 증적소에 저장될 수 있도록 포맷에 맞게 감사 기록 생성	실행된 스레드의 id를 식별한 후, 해당 스레드의 실행 내역(함수 명, 실행 여부)을 audit_data 구조체에 저장	
	audit_check_write_authorize	*tp: 스레드 포인터	해당 스레드가 감사 기록을 생성하는 것이 적절한지 점검	현재 감사 기록을 생성하려고 한 스레드의 식별자 값이 ch_auditbox.wr_thread 내에 저장되었는지 비교	
	audit_register_id	_id: 스레드 식별자	감사 기록을 생성할 수 있는 스레드 식별자 저장	ch_auditbox.wr_thread에 입력 매개 변수 값 갱신	
	audit_remove_id	_id: 스레드 식별자	감사 기록을 생성할 수 있는 스레드 식별자 목록 중 입력 매개변수 값 삭제	ch_auditbox.wr_thread에 입력 매개 변수 값 제거	
	auditbox_update_buffer	audit_data: 감사 기록	공유되는 메모리 공간에 감사 기록 저장 및 관련 플래그 설정	ch_auditbox.buffer에 입력 매개 변수 값을 저장하고, 커널 스레드가 접근할 수 있도록 ch_auditbox.flag를 True로 변환	
	audit_increment	-	이후 감사 기록이 증적될 메모리 위치 업데이트	감사 기록이 증적된 후, 이후 감사 기록이 증적될 메모리 위치 업데이트	
	auditspace_is_memory_full	-	현재 감사 증적소의 메모리 잔여량 계산	현재 감사 증적소의 메모리 잔여량을 계산한 후, 꽉 차는 경우 감사 기록을 덮어 쓸 수 있도록 True를 반환하고, 그렇지 않을 경우 False 반환	
	auditspace_read_data_from_auditbox	-	감사 기록을 감사 증적소에 저장하려는 스레드가 커널 권한을 가졌는지 점검	현재 쓰기를 수행하려는 스레드가 ch_auditbox.re_thread 값과 동일한지 비교	
	auditspace_copy_buffer_from_auditbox		감사 증적소에 감사 기록 증적	감사 증적소의 메모리 공간인 ch_auditspace.space에 공유되는 메모리 공간에 저장된 ch_auditbox.buffer 값을 저장	

[표 3] 감사 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능 요구사항을 만족한다.

TSFI \ SFR	FAU_GEN.1	FAU_STG.1	FAU_STG.3	FAU_STG.4
audit_try_update_buffer	X			
audit_check_write_authorize		X		
audit_register_id		X		
audit_remove_id		X		
auditbox_update_buffer	X			
audit_increment				
auditspace_is_memory_full			X	X
auditspace_read_data_from_auditbox	X			
auditspace_copy_buffer_from_auditbox			X	X

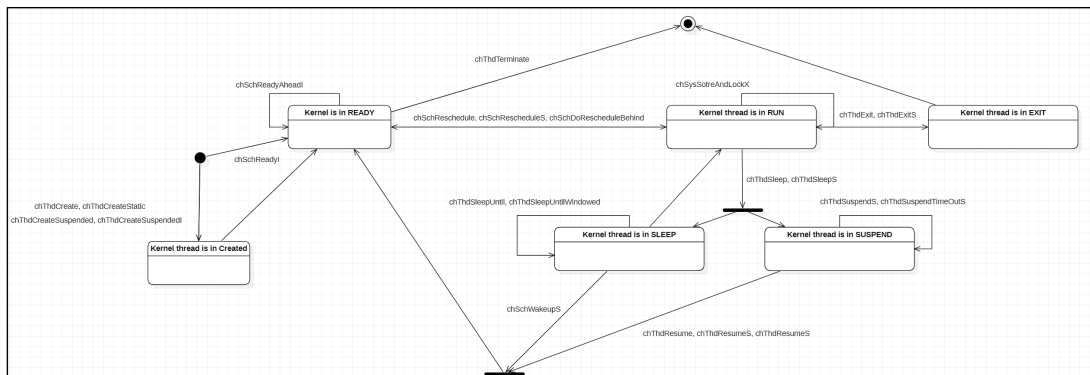
[표 4] 감사 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스

3.2. FRU(자원 활용) 관련 TSFI

본 절에서는 자원 활용 관련 TSFI에 대해 서술한다. 자원 활용 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.

사용자의 요청과 TOE의 내부 상태가 모두 정상일 경우, TOE의 FRU 서비스시스템이 트리거된다. 해당 서비스시스템은 사용자의 요청을 우선순위에 따라 차례대로 처리한다. 사용자가 신규로 특정 작업을 요청하는 경우 입력받은 매개변수에 따라 TOE는 chThdCreate, chThdCreateStatic, chThdCreateSuspended, chThdCreateSus[emded] 모듈을 통해 필요한 자원을 할당받을 때 까지 대기한다. 만약 해당 작업이 과거에 처리한 작업을 반복하는 경우라면, chSchReadyI, chSchReadyAheadI 모듈이 실행된다. 이후 대기중인 사용자 작업을 처리하기 위해 chSchReschedule, chSchRescheduleS, chSchDoRescheduleBehind가 실행된다. 이후 사용자가 더 처리할 작업이 없는 경우, TOE는 chThdTerminate, chThdExit, chThdExitS 모듈을 통해 관련 자원을 회수하고 관련 작업을 종료시킨다.

TOE는 사용자가 다양한 작업을 특정 순서에 따라 처리할 수 있도록 chThdSleep, chThdSleepS 모듈을 통해 실행 중인 사용자 작업을 일시 중지시킨다. 이후 선 처리해야 하는 작업을 수행하기 위해 사전에 대기중인 작업을 chSchWakeups, chThdResume, chThdResumeS, chThdResumeI 모듈을 통해 실행한다.



[그림 10] 자원 활용 모듈에 의한 TOE 세부 상태전이도

자원 활용 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개 변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	chSchReadyI	k V a l u e : 입력받은 비밀값 tp: 레디 리 스 트 에 추가될 스레드 포인터	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	매개변수로 전달된 비밀값이 커널의 비밀값과 같을 때만 함수가 실행. 매개변수로 전달된 스레드 포인터를 레디 상태로 변경하고 레디 리스트의 제일 뒤에 삽입	

SFR 수행	chSchReady AheadI	kvalue 입력받은 비밀값 tp : 레디 리 스토에 추가될 스레드 포인터	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	매개변수로 전달된 비밀값이 커널의 비밀값과 같을 때만 함수가 실행된다. 매개변수로 전달된 스레드 포인터를 레디 상태로 변경하고 레디 리스트의 제일 앞에 삽입	
	chSchGoSleepS	newstate : 스레드에 게 새로 할당될 상태	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	현재 실행 중인 스레드의 동작을 일시 정지시키고 매개변수로 주어진 새로운 상태로 변환.	
	chWakeupS	-	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	인터럽트 서비스 루틴에 잠시 락을 걸고, 해당 스레드의 상태에 맞는 동작들을 수행한 다음, 해당 스레드를 레디 상태로 전환하고, 인터럽트 서비스 루틴의 락을 해제	
	chSchGoSleep TimeoutS	newstate : 스레드에 게 새로 할당될 상태 timeout : 스레드의 실행이 일시 정지될 시간	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	현재 실행 중인 스레드의 동작을 일시 정지시키고 매개변수로 주어진 새로운 상태로 변환. 이때, 매개변수로 주어진 틱 수만큼만 일시정지	
	chSchRescheduleS	-	높은 우선순위를 가지는 사용자 요청 처리	만약 현재 스레드보다 높은 우선순위를 갖는 실행 가능한 스레드가 레디 리스트에 있는 경우, 해당 스레드가 실행	
	chSchIsPreemption Required	-	높은 우선순위를 가지는 사용자 요청 처리	현재 실행 중인 스레드의 우선순위와 레디 리스트에서 대기 중인 스레드의 최고 우선순위를 비교	
	chSchDoReschedule Behind	-	높은 우선순위를 가지는 사용자 요청 처리	현재 실행 중인 스레드의 포인터를 레디 리스트 맨 뒤에 삽입 후, 레디 리스트 내 가장 높은 우선순위를 가진 스레드를 실행	
	chSchDoReschedule Ahead	-	높은 우선순위를 가지는 사용자 요청 처리	현재 실행 중인 스레드의 포인터를 레디 리스트 맨 앞에 삽입 후, 레디 리스트 내 가장 높은 우선순위를 가진 스레드를 실행	
	chSchDoReschedule	-	높은 우선순위를 가지는 사용자 요청 처리	현재 실행 중인 스레드의 포인터를 해당 스레드에게 할당된 시간을 모두 사용했는지에 따라 레디 리스트의 맨 뒤나 맨 앞에 삽입 후, 레디 리스트 내 가장 높은 우선순위를 가진 스레드를 실행	

SFR 수행	chThdWait	tp: 스레드 포인터	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	매개변수로 주어진 스레드가 종료되어 종료 코드가 반환될 때까지 이 함수를 호출한 스레드의 실행을 일시 중지하고 대기	
	chThdSleepUntil	time: 절대 시스템 시간	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	시스템 시간이 매개변수로 주어진 시간에 도달할 때까지, 이 함수를 호출한 스레드의 실행을 유예	
	chThdSleepS	ticks: 스레드가 일시 정지될 시간 만큼의 틱 수	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	매개변수로 주어진 틱 수만큼 스레드의 실행을 유예	
	chThdSleepUntil Windowed	prev: 일시 정지가 시작될 시간 next: 일시 정지가 끝날 시간	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	시스템 시간이 매개변수로 주어진 시간 사이에 있으면, 이 함수를 호출한 스레드의 실행을 일시 정지	
	chThdStartI	k V a l u e : 입력받은 비밀값 tp: 입력 받은 스레드 포인터	높은 우선순위를 가지는 사용자 요청 처리	매개변수로 입력받은 스레드를 다시 시작	
	chThdSleep	m s e c : 스레드를 일시 정지할 시간	의도치 않게 우선순위가 높은 사용자 요청을 처리하기 위해 현재 처리 중인 사용자 요청 처리 일시 중지	매개변수로 주어진 틱만큼 스레드를 일시 정지.	
	chThdResume	k V a l u e : 입력받은 비밀값 trp: 스레드 참조 객체에 대한 포인터 msg: 메시지 코드	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	매개변수로 주어진 스레드 참조 객체에 대해 대기중이던 스레드를 깨움	
	chThdResumeI	k V a l u e : 입력받은 비밀값 trp: 스레드 참조 객체 포인터 msg: 메시지 코드	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	매개변수로 주어진 스레드 참조 객체에 대해 대기중이던 스레드를 깨움	
	chThdResumeS	k V a l u e : 입력받은 비밀값 trp: 스레드 참조 객체 포인터 msg: 메시지 코드	사용자가 우선순위에 따라 CPU를 할당받을 수 있도록 사용자 요청을 대기열에 추가	매개변수로 주어진 스레드 참조 객체에 대해 대기 중이던 스레드를 깨움	
	chThdCreatel	k V a l u e : 입력받은 비밀값 tdp: 스레드 디스크립터	사용자에게 고정된 메모리 크기를 할당	매개변수로 주어진 스레드 디스크립터를 바탕으로 정적 메모리 영역에 새로운 스레드를 생성	
	chThdCreateStatic	k V a l u e : 입력받은 비밀값 tdp: 스레드 디스크립터	사용자에게 고정된 메모리 크기를 할당	매개변수로 주어진 스레드 디스크립터를 바탕으로 정적 메모리 영역에 새로운 스레드를 생성	
	chThdTerminate	tp: 스레드 포인터	사용자에게 할당되었던 메모리 회수	매개변수로 주어진 스레드 포인터에게 종료 요청	

SFR 수행	chThdRelease	tp: 스레드 포인터	사용자에게 할당되었던 메모리 회수	매개변수로 주어진 스레드 포인터가 가리키는 스레드 객체를 메모리에서 해제	
	chThdExit	k V a l u e : 입력받은 비밀값 msg: 스레드 종료 코드	사용자에게 할당되었던 메모리 회수	현재 실행 중인 스레드를 종료	
	chThdCreateSuspendedI	k V a l u e : 입력받은 비밀값 tp: 입력받은 스레드 정보	사용자에게 고정된 메모리 크기를 할당	매개변수로 주어진 스레드 정보를 바탕으로 정적 메모리 영역에 새로운 스레드를 생성. 새 스레드는 초기화되지만 레디 리스트에 삽입되지는 않음	
	chThdCreateSuspended	k V a l u e : 입력받은 비밀값 tp: 입력받은 스레드 정보	사용자에게 고정된 메모리 크기를 할당	매개변수로 주어진 스레드 정보를 바탕으로 정적 메모리 영역에 새로운 스레드를 생성. 새 스레드는 초기화되지만 레디 리스트에 삽입되지는 않음	

[표 5] 자원 활용 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능
요구사항을 만족한다.

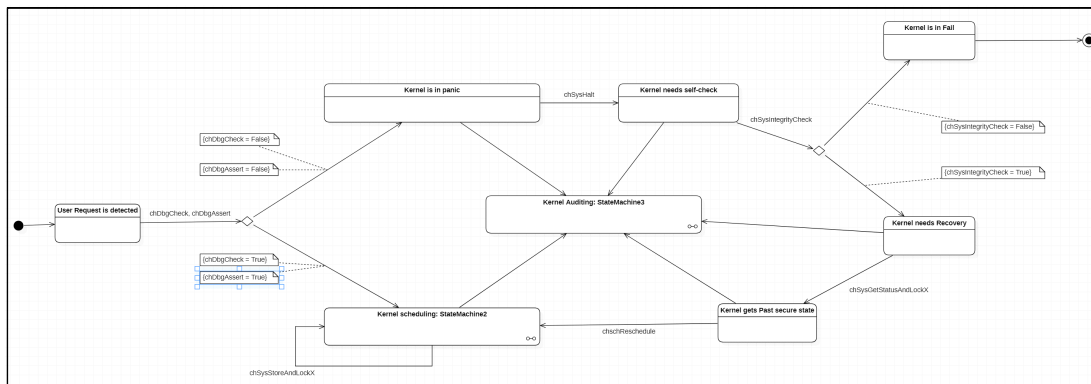
TSFI \ SFR	FRU_PRS.1	FRU_RSA.1
chSchReadyI	X	
chSchReadyAheadI	X	
chSchGoSleepS	X	
chWakeupS	X	
chSchGoSleepTimeoutS	X	
chSchRescheduleS	X	
chSchIsPreemptionRequired	X	
chSchDoRescheduleBehind	X	
chSchDoRescheduleAhead	X	
chSchDoReschedule	X	
chThdWait	X	
chThdSleepUntil	X	
chThdSleepS	X	
chThdSleepUntilWindowed	X	
chThdStartI	X	
chThdSleep	X	
chThdResume	X	
chThdResumeI	X	
chThdResumeS	X	
chThdCreateI		X
chThdCreateStatic		X
chThdTerminate		X
chThdRelease		X
chThdExit		X
chThdCreateSuspendedI		X
chThdCreateSuspended		X

[표 6] 자원 활용 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스

3.3. FPT(자체 보호) 관련 TSFI

본 절에서는 자체 보호 관련 TSFI에 대해 서술한다. 자체 보호 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.

사용자 스레드로부터 TOE에 요청이 오면, FPT 서브시스템이 트리거된다. 해당 서브시스템은 사용자 요청의 입력 값이 유효 범위 안에 있는지 chDbgCheck와 chDbgAssert 모듈을 통해 점검한다. 모듈의 반환 값 중 하나라도 False일 경우, TOE는 Syshalt 모듈을 통해 커널의 동작을 중지한 후, chSysIntegrityCheck 모듈을 활용하여 내부 상태를 점검한다. 만약 chSysIntegrityCheck 모듈의 반환 값이 True인 경우, TOE는 chSysGetStatusAndLockX 모듈을 활용하여 이전의 안전한 상태로 복구된다. 안전한 상태는 스케줄링 관련 모듈(관련 서브시스템: FRU)이 리스케줄될 때 마다, chSysStoreandLockX 함수를 통해 저장된다. 이때 모든 커널의 동작은 감사 모듈(관련 서브시스템: FAU)에 의해 기록된다.



[그림 11] 자체 보호 모듈에 의한 TOE 상태전이도

자체 보호 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	chDbgCheck	c: 점검해야 하는 조건	입력 매개변수 유효성 점검	입력 매개변수 유효성 점검	
	chDbgAssert	c: 점검해야 하는 조건 r: 준수해야 하는 조건	모듈 내 논리적 모순 점검	모듈 실행 시, TOE의 논리적 모순 발생 여부 점검	
	chSysGetStatusAndLockX	-	오류 발생 시, 과거 안전한 상태로 복구	시스템에 락을 걸고, 임계구역에 진입한 후, 현재 실행 상태를 반환	
	chSysRestoreStatusX	sts: 입력받은 실행 상태	오류 발생 시, 과거 안전한 상태로 복구될 수 있는 체크포인트 저장	매개변수로 주어진 실행 상태로 복구하고, 임계 구역에서 나온 후, 락을 해제	
	chSysIntegrityCheck		커널 상태 점검	커널 내 자료구조 내 링크드 리스트 내 포인터 내 모순이 없는지 여부에 대해 자체 점검	

[표 7] 자체 보호 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능 요구사항을 만족한다.

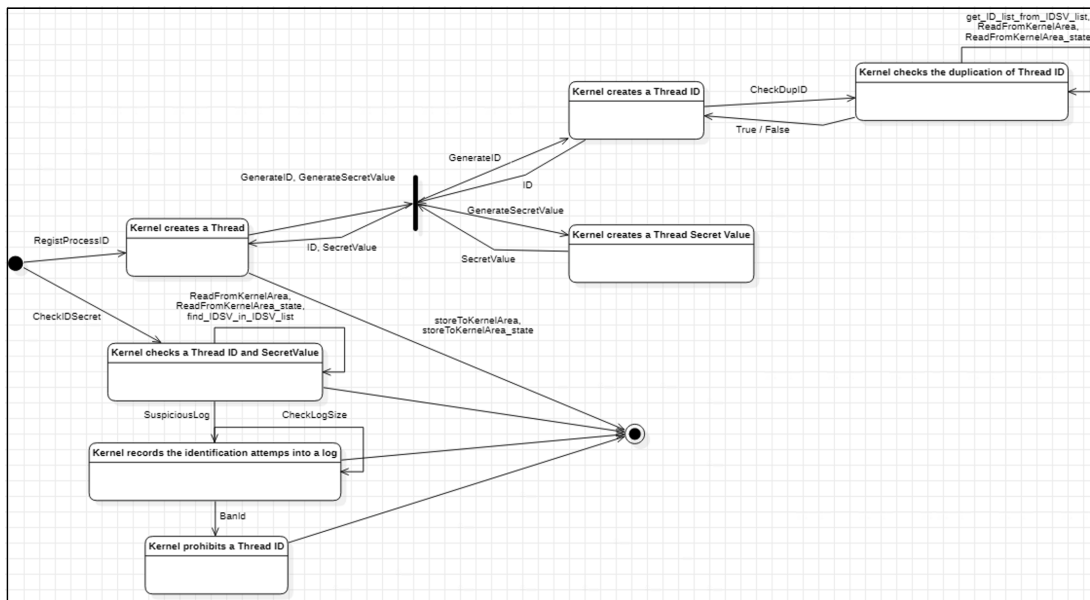
TSFI \ SFR	FPT_FLS.1	FPT_RCV.1	FPT_TST.1
chDbgCheck	X		X
chDbgAssert	X		X
chSysGetStatus AndLockX		X	
chSysRestore StatusX	X		
chSysIntegrity Check	X		X

[표 8] 자체 보호 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스

3.4. FIA(식별 및 인증) 관련 TSFI

본 절에서는 식별 및 인증 관련 TSFI에 대해 서술한다. 식별 및 인증 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.

TOE는 사용자의 요청에 따라 서로 다른 식별 및 인증 관련 TSFI가 수행된다. 사용자가 스레드를 생성하는 경우, `RegistProcessID`, `GenerateID`, `GenerateSecretValue` 모듈을 통해 스레드의 ID 및 Secret Value를 생성한 뒤, 커널 영역의 저장소에 저장한다. 스레드 ID의 중복성을 제거하기 위해, `get_ID_list_from_IDS_V_list`, `ReadFromKernelArea`, `ReadFromKernelArea_state` 모듈로 이미 생성된 스레드 ID와 비교하여 중복성을 확인한다. 생성된 스레드 ID와 Secret Value는 `storeToKernelArea`, `storeToKernelArea_state` 모듈을 통해 커널 영역에 저장된다. 사용자가 스레드의 ID와 Secret Value의 인증을 원하는 경우, `CheckIDSecret` 모듈을 통해 스레드 ID와 Secret Value를 인증할 수 있다. `ReadFromKernelArea`, `ReadFromKernelArea_state`, `find_IDS_V_in_IDS_V_list` 모듈은 커널 영역에 저장된 스레드 ID와 Secret Value를 읽어서 `CheckIDSecret` 모듈을 통해 전달받은 스레드 ID와 Secret Value와 비교 기능을 지원한다. 만약 인증에 실패하는 경우, `SuspiciousLog` 모듈을 통해 로그 파일에 해당 인증 시도를 기록한다. 로그에 기록을 할 때에는 `CheckLogSize` 모듈을 통해 로그 파일의 오버플로우가 발생하는 것을 방지한다. 일정 임계값(예: 5) 이상의 인증이 실패한 경우, `BanID` 모듈을 통해 스레드의 ID와 Secret Value를 비활성화한다.



[그림 12] 식별 및 인증 모듈에 의한 TOE 세부 상태전이도

식별 및 인증 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	GenerateID	SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체만 ID 할당	입력받은 비밀 값이 커널의 비밀 값과 같다면, ID를 생성 후 반환	
	GenerateSecret Value	SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체만 비밀 값 할당	입력받은 비밀 값이 커널의 비밀 값과 같다면, 비밀 값을 생성 후 반환	
	get_ID_list_from_IDSV_list	IDSV_list_t: ID와 비밀 값의 쌍으로 이루어진 리스트 ID list: ID로 이루어진 리스트	IDSV 쌍으로부터 ID만으로 이루어진 리스트를 반환하여 비밀 값은 노출하지 않고 ID만 반환	ID와 비밀 값의 쌍으로 이루어진 리스트를 입력받아 ID만 뽑아 새로운 리스트를 만들어서 반환	
	get_SV_list_from_IDSV_list	IDSV_list_t: ID와 비밀 값의 쌍으로 이루어진 리스트 SecretValue list: 비밀 값으로 이루어진 리스트	IDSV 쌍으로부터 비밀 값만으로 이루어진 리스트를 생성 후 반환	ID와 비밀 값의 쌍으로 이루어진 리스트를 입력받아 비밀 값만 뽑아 새로운 리스트를 만들어서 반환	
	CheckDuplID	ID: 입력받은 ID SecretValue: 입력 받은 비밀 값	커널의 비밀 값을 가진 객체의 ID생성 시 중복 방지	입력받은 비밀 값이 커널의 비밀 값이 맞다면 입력받은 ID가 등록된 ID와 중복되는지 확인	
	storeToKernel Area_state	SecretValue: 입력받은 비밀 값 ID: 입력받은 ID SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체의 ID와 비밀 값을 커널영역의 변수에 저장	입력받은 비밀 값이 커널의 비밀 값이라면, 입력받은 ID와 비밀 값을 커널 영역의 변수인 IDSV_list에 추가	
	storeToKernel Area	SecretValue: 입력받은 비밀 값 ID: 입력받은 ID SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체만이 storeToKernelArea_state 함수를 호출	입력받은 비밀 값이 커널의 비밀 값이라면, storeToKernelArea_state 함수를 호출	
	ReadFromKernel Area_state	SecretValue: 입력받은 비밀 값 Nat: 색인번호	커널의 비밀 값을 가진 객체만이 Index를 활용하여 ID와 비밀 값 셋 반환	입력받은 비밀 값이 커널의 비밀 값이라면, IDSV_list에서 입력받은 Index에 있는 ID와 비밀 값 셋을 반환	

SFR 수행	ReadFromKernel Area	SecretValue: 입력받은 비밀 값 Nat: 색인번호	커널의 비밀 값을 가진 객체만이 ReadFromKernel Area_state 함수를 호출	입력받은 비밀 값이 커널의 비밀 값이라면, ReadFromKernel Area_state 함수를 호출	
	RegistProcessID	SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체만 ID와 비밀값 생성 및 반환	입력받은 비밀 값이 커널의 비밀 값이면 ID와 비밀 값을 생성하고 커널 영역에 생성된 ID와 비밀 값을 저장한 뒤 ID와 비밀 값을 반환	
	find_IDS_V_in_IDS_V_list	SecretValue: 입력받은 비밀 값 ID 입력받은 ID SecretValue: 입력받은 비밀 값 IDS_V_list: ID와 비밀 값의 쌍으로 이루어진 리스트	커널의 비밀 값을 가진 객체만 ID와 비밀 값 쌍의 리스트에서 ID와 비밀 값의 존재여부 검색 및 반환	입력받은 비밀 값이 커널의 비밀 값이라면, ID와 비밀 값의 쌍으로 이루어진 리스트에서 입력받은 ID와 비밀 값이 존재하는 여부에 대해 결과를 반환	
	CheckIDSecret	SecretValue: 입력받은 비밀 값 ID: 입력받은 ID SecretValue: 입력받은 비밀 값	커널의 비밀 값을 가진 객체만 IDS_V_list에 ID와 비밀 값 쌍이 존재하는지 판별하는 find_IDS_V_in_IDS_V_list 호출	입력받은 ID와 비밀 값이 IDS_V_list에 존재하는지 여부를 확인	
	BanId	SecretValue: 입력받은 비밀 값 ID: 입력받은 ID SecretValue: 입력받은 비밀 값 timestamp: 현재 타임스탬프	커널의 비밀 값을 가진 객체만 ban_list를 통해 ID와 비밀 값, 현재 타임스탬프를 추가	입력받은 비밀 값이 커널의 비밀 값이라면, ban_list에 입력받은 ID와 비밀 값, 현재 타임스탬프를 추가	
	CheckLogSize	Log: 로그	로그 파일의 크기 점검	로그 파일의 크기 점검	
	SuspiciousLog	SecretValue: 입력받은 비밀 값 ID: 입력받은 ID SecretValue: 입력받은 비밀 값 timestamp: 현재 타임스탬프 Nat: 로그인 시도 회수 Log: 로그	커널의 비밀 값을 가진 객체의 로그인 시도 회수가 일정 기준을 넘지 않으면 로그 파일에 로그인 시도 로그를 기록하고, 기준을 넘으면 BanId를 호출	로그인 시도 회수가 일정 기준을 넘지 않으면 로그 파일에 로그인 시도 로그를 기록하고, 기준을 넘으면 BanId를 호출	

[표 9] 식별 및 인증 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능 요구사항을 만족한다.

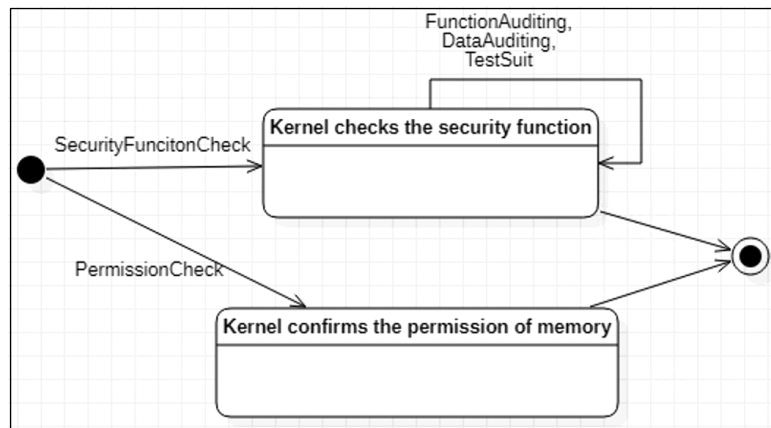
TSFI \ SFR	FIA_UID.1	FIA_UAU.1
GenerateID	X	
GenerateSecretValue	X	
get_ID_list_from_IDSV_list	X	
get_SV_list_from_IDSV_list	X	
CheckDupID	X	
storeToKernelArea_state	X	
storeToKernelArea	X	
ReadFromKernelArea_state	X	
ReadFromKernelArea	X	
RegistProcessID	X	
find_IDSV_in_IDSV_list		X
CheckIDSecret		X
BanId		X
CheckLogSize		X
SuspiciousLog		X

[표 10] 식별 및 인증 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스

3.5. FMT(보안 속성 관리) 관련 TSFI

본 절에서는 보안 속성 관리 관련 TSFI에 대해 서술한다. 보안 속성 관리 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.

사용자가 보안 기능의 무결성을 확인하고자 할 때에는, SecurityFunctionCheck 모듈을 통해 함수의 보안기능의 무결성을 확인할 수 있다. 또한, FunctionAuditing, DataAuditing 모듈을 통해 보안기능을 갖는 함수의 동작을 기록하고 데이터의 복제, 전송 등의 행동이 권한 내에서 이루어지는 확인한다. TestSuit 모듈을 통해 미리 정의된 테스트 케이스로 보안성을 확인한다. 사용자가 메모리 주소의 권한을 확인하고자 하는 경우, PermissionCheck 모듈을 통해 해당 메모리 주소의 권한이 커널 권한을 갖는지 확인한다.



[그림 13] FMT 모듈에 의한 TOE 세부 상태전이도

보안 속성 관리 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	SecurityFunction Check	fname_in: 확인하고자 하는 함수명	입력된 보안기능을 확인	보안 기능을 확인하고자 하는 함수의 이름을 넣으면 결과를 반환	
	FunctionAuditing	fname_in: 확인하고자 하는 함수의 명 action: 함수의 행동	보안기능을 확인하고자 하는 함수의 행동을 기록	함수의 행동을 로그 파일로 기록	
	DataAuditing	data_name: 데이터 이름 data_behavior: 데이터의 행동	데이터의 이름과 행동을 입력받아 허가된 데이터 행동에 포함되는지 판별	데이터의 이름과 데이터의 행동을 입력받아 데이터 이름이 허가된 데이터 행동에 포함되는지 여부를 반환	
	TestSuit	name: 테스트 수트의 이름 test: 테스트 수트	테스트 수트를 입력받아 미리 선정한 테스트 수트에 포함되는지 여부 판별	테스트 수트를 입력받아 미리 선정한 테스트 수트에 포함되는지 여부를 반환	
	PermissionCheck	addr_in: 권한을 확인할 메모리 주소	메모리 주소를 입력받아 해당 주소의 권한이 적합한지 확인	메모리 주소를 입력받아 해당 주소의 권한이 적합한지 확인	

[표 11] 보안 속성 관리 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능 요구사항을 만족한다.

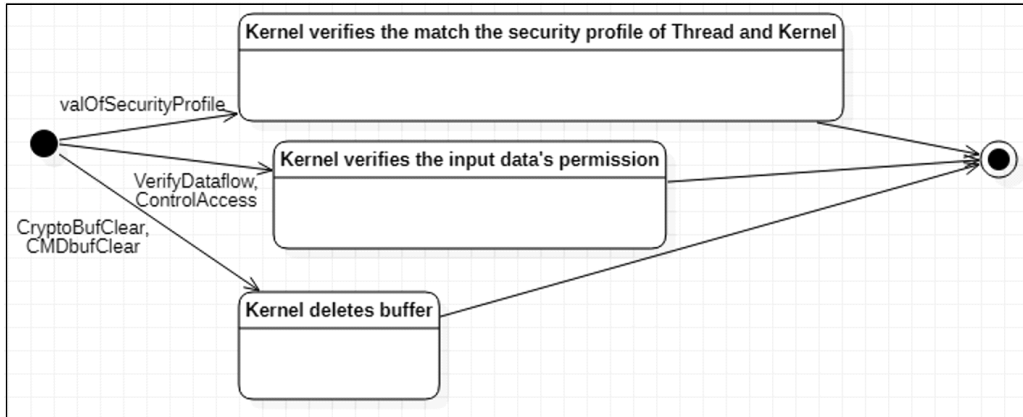
TSFI	SFR					
	FMT_MOF.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1
SecurityFunctionCheck	X				X	X
FunctionAuditing		X		X	X	X
DataAuditing		X			X	X
TestSuit				X	X	
PermissionCheck	X	X	X	X		

[표 12] 보안 속성 관리 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스

3.6. FDP(사용자 데이터 보호) 관련 TSFI

본 절에서는 사용자 데이터 보호 관련 TSFI에 대해 서술한다. 사용자 데이터 보호 관련 TSFI에 의해 TOE의 내부 상태는 아래 상태전이도에 따라 변경된다.

사용자는 valOfSecurityProfile 모듈을 통해 쓰레드의 보안 속성이 커널의 보안 속성과 일치하는지 확인할 수 있다. 데이터와 명령어의 권한의 확인은 VerifyDataflow, ControlAccess 모듈을 통해 수행된다. 데이터 유출을 방지하기 위해서 CryptoBufClear, CMDbufClear 모듈을 통해 암호화에 사용되거나 명령어에 사용된 버퍼의 데이터를 초기화한다.



[그림 14] FDP 모듈에 의한 TOE 세부 상태전이도

보안 속성 관리 관련 TSFI의 세부 정보는 TSFI 명, 매개변수 정보, TSFI 수행 목적, 수행 동작(호출 조건), 오류 메시지가 포함된다.

구분	TSFI 명	매개변수 정보	수행 목적	수행 동작(호출 조건)	오류 메시지
SFR 수행	valOfSecurityProfile	subjectProfile: 검사하고 싶은 보안속성	주체의 보안속성과 커널의 보안속성이 일치하는지 점검	검사하고자 하는 보안속성을 입력받아 커널에 저장된 보안속성과 일치하는지 판별	
	VerifyDataflow	data: 접근되는 데이터 adress: 메모리 주소	접근하고자 하는 데이터와 주소의 권한을 비교하여 일치 여부를 판별	접근되는 데이터와 메모리 주소를 입력받아 권한을 확인 후 일치하는지 판별	
	ControlAccess	instruction: 수행 동작 thread_t: 스레드	수행동작과 해당 스레드의 권한여부를 판별하여 권한을 통제	수행동작과 해당 스레드를 입력으로 받아 수행동작의 권한과 스레드의 권한을 확인 후 일치하는지 판별	
	CryptoBufClear	-	암호화에 사용된 버퍼의 크기만큼 초기화하여 잔여 정보 유출 방지	암호화에 사용된 버퍼의 크기만큼 초기화	
	CMDbufClear	-	수행된 동작에 사용된 버퍼의 크기만큼 초기화하여 잔여 정보 유출 방지	수행된 동작에 사용된 버퍼의 크기만큼 초기화	

[표 13] FDP 관련 TSFI 목록

상기 TSFI의 수행 목적과 동작을 근거로 아래와 같이 보안기능이 보안기능

요구사항을 만족한다.

TSFI	SFR	FDP_IFC.1	FDP_IFF.1
valOfSecurityProfile		X	
VerifyDataflow			X
ControlAccess			X
CryptoBufClear			
CMDbufClear			

[표 14] FDP 클래스 보안기능 요구사항과 TSFI 간 추적성 매트릭스