<부록4. 보안 아키텍쳐 문서>

CHAOS ver1.0
보안 아키텍쳐 문서 ver1.0

(ADV_ARC.1)

- 고등급 보안 마이크로커널 개발 -

<목 차>

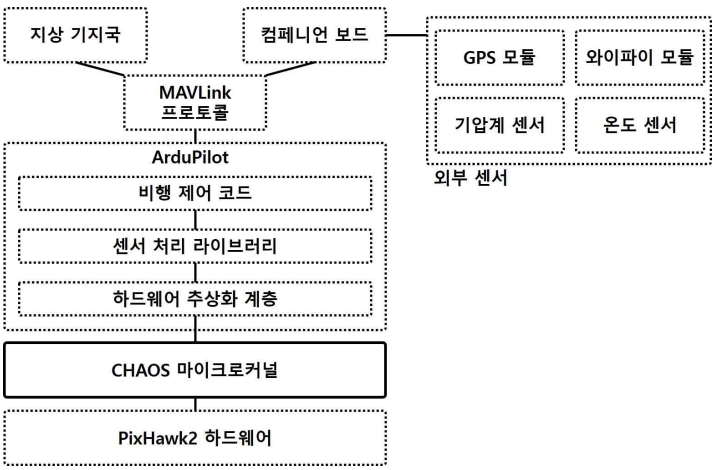
1. 개요

본 문서는 TOE에 해당하는 CHAOS(Chibi-bsed High Assurance Operating System)의 보안기능이 어떠한 구조로 구성되어 있는지 설명하는 보안 아키텍쳐 문서(ADV_ARC.1)이다. 보안 제품 개발 단계에서 산출되는 보안 아키텍쳐 문서(ADV_ARC.1)에서는 두가지 특성을 입증해야 한다. 첫 번째 특성은 보안 기능이 명세된 대로 동작해야 한다는 정확성이고, 두 번째 특성은 보안 기능이 손상되거나 우회될 수 없어야 한다는 우회불가성 이다. 두 번째 특성의 경우, 기능 손실이나 관련 위협의 모든 가능성을 고려해야 하므로 첫 번째 특성보다 입증하기가 어렵다.

아키텍쳐 문서에서는 자체 보호, 우회 불가성 총 2가지의 특성을 입증해야 한다. 자체 보호란 보안 기능에 변경을 일으킬 수 있는 외부 실체의 조작으로부터 자신을 보호하는 능력을 의미하며, 보안기능이 항상 정상적으로 동작함을 입증해야 한다.

우회불가성이란 권한이 없는 주체가 우회경로를 통해 권한이 요구되는 영역에 접근하지 못하도록 우회경로를 차단하는 것을 의미하며, 우회불가성은 권한 식별 및 인증 과정을 통해 주체에 대한 인증 과정 없이는 해당 영역에 접근할 수 없도록 해야 한다. 외부 주체가 TSF를 우회하여 자산에 접근할 수 있는 경로가 모두 차단됨을 입증해야 한다.

TOE는 UAV(Unmanned Aerial Vehicle)의 한 종류인 콥터(Copter), 즉, 드론에 탑재되는 마이크로커널이다. 다음 [그림 2]는 TOE가 탑재된 드론의 전체 구성도를 보여준다.



[그림 1] 드론 시스템의 구성도

1장은 본 문서의 개요와 목적을 보여주며 2장에서는 전체 드론 구성 중 본 문서에서 다루는 TOE에 대해 보안기능과 비 보안기능이 어떻게 구성되어 있는지 보여준다. 마지막으로 3장에서는 2장에서 설명한 마이크로커널이 2가지 보안구조에 대해 어떻게 달성하고 있는지 상세하게 설명한다.

1.1. 문서의 목적

TOE는 보안기능이 탑재되어 있지 않은 마이크로커널인 ChibiOS를 기반으로

보안기능을 추가한 마이크로커널이다. 본 문서는 TOE에 대한 보안 아키텍쳐를 설명하는 것으로 공통평가기준의 EAL6수준을 만족하기 위해 ADV_ARC.1 컴포넌트에서 요구하는 증거 요구사항에 맞게 2가지 보안구조(자체보호, 우회 불가성)에 대해 설명한다.

1.2. 시스템의 전체 구조와 평가 범위

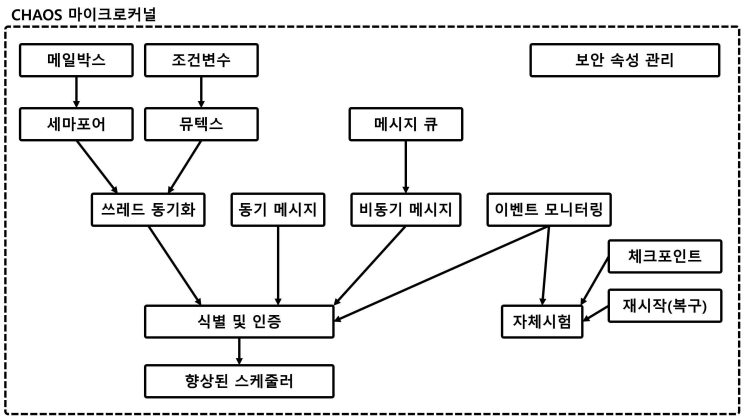
1.2.1. 전체 구조

본 절에서는 위의 [그림 1]에서 대략적으로 보여준 시스템의 각 구성요소에 대해 상세하게 설명한다. 먼저 전체 드론시스템에서 프로세서 역할을 하는 하드웨어는 PixHawk2를 사용한다. 해당 장비의 경우 GPS(Global Positioning System) 연결 포트를 비롯하여 기타 장비들과 유선통신을 할 수 있는 I2C(Inter Integrated Circuit), CAN(Controller Area Network) 통신 포트를 지원하며 무선 통신을 할 수 있는 포트로 무선 조종기의 입력을 받아들이는 RC-in(Remote Control in) 포트, MAVLink와 같은 프로토콜을 사용하여 무선통신하기 위한 Telemetry 포트도 지원한다.

PixHawk2 하드웨어를 제어하기 위해 탑재되는 펌웨어는 두 부분으로 구분되는데 첫 번째 부분은 자원을 효과적으로 관리하기 위한 TOE 마이크로커널이고, 두 번째 부분은 실제 비행 제어를 수행하기 위한 라이브러리인 ArduPilot이다. 드론의 단순한 비행을 위해선 TOE없이 ArduPilot에서 기본으로 제공하는 운영체제인 NuttX 만으로 비행할 수 있지만, 드론이 실제 주어진 임무(탐색, 지상 센서 노드로부터 데이터 수집, 편대 비행 등)를 수행하기 위해 많은 작업이 추가될 경우 NuttX 운영체제의 높은 복잡성으로 인해 효율성이 떨어지게 된다. 이에 본 시스템에선 자원을 더욱 효율적으로 관리하기 위해 ChibiOS 기반의 TOE를 탑재하였다.

1.2.2. 평가 범위

이전 절에서 설명한 전체 범위 중 본 연구팀이 보안기능을 추가하고 평가 받고자 하는 범위는 위의 [그림 1]에서 실선으로 표기된 부분과 같이 드론의 메인 컴퓨터에 탑재되는 CHAOS 보안 마이크로커널이다. TOE는 기존에 개발되어있는 마이크로커널인 ChibiOS를 기반으로 보안 기능을 추가하여 개발한 것으로 ChibiOS에 대해 체계적인 위협 분석 방법론을 적용하여 전체 위협을 도출한 후 각 위협에 대한 완화방안을 포함시켰다. 다음 [그림 2]는 TOE의 전체 구성도를 나타낸다.



[그림 2] 3차연도 TOE 전체 구성도

2. 보안기능의 개요

본 장에서는 TOE의 보안기능과 해당 보안기능이 완화하고자 하는 위협에 대해 설명한다. TOE는 [그림 2]와 같이 스레드 동기화, 동기 메시지, 비동기 메시지, 이벤트, 향상된 스케줄링, 자체보호기능이 있으며 전체 기능 중 보안기능은 크게 1)향상된 스케줄링 기능, 2)자체보호기능, 3)식별 및 인증기능 4)보안 속성 관리 기능이 존재하고, 비 보안기능으로 스레드 동기화, 동기 메시지, 비동기 메시지기능이 존재한다.

본 마이크로커널은 이전에 설명된 바와 같이 ChibiOS를 기반으로 보안성을 향상시켜서 개발되었다. 이를 위해 기존 ChibiOS에 대해 체계적으로 도출된 전체 위협과 이를 완화시키기 위한 완화방안을 도출하였다. 다음 [표 1]은 도출된 위협과 해당 위협을 완화시킬 수 있는 보안기능 요구사항을 서로 매핑한 결과를 보여준다.

위협	보안기능 요구사항
안전하지 않은 상태	안전한 상태 유지(FPT_FLS.1)
	자동 복구(FPT_RCV.2)
	TSF 자체 시험(FPT_TST.1)
	보안기능 관리(FMT_MOF.1)
	보안속성 관리(FMT_MSA.1)
	정적 속성 초기화(FMT_MSA.3)
	TSF 데이터 관리(FMT_MTD.1)
	관리기능 명세(FMT_SMF.1)
	보안 역할(FMT_SMR.1)
TSF 우회	인증(FIA_UAU.1)
	식별(FIA_UID.1)
	보안기능 관리(FMT_MOF.1)
	보안속성 관리(FMT_MSA.1)
	정적 속성 초기화(FMT_MSA.3)
	TSF 데이터 관리(FMT_MTD.1)
	관리기능 명세(FMT_SMF.1)
	보안 역할(FMT_SMR.1)
서비스 거부	자원사용 우선순위: 부분적용(FRU_PRS.1)
	최대 할당치 제한(FRU_RSA.1)
TSF 데이터 유출	인증(FIA_UAU.1)
	식별(FIA_UID.1)
	보안기능 관리(FMT_MOF.1)
	보안속성 관리(FMT_MSA.1)
	정적 속성 초기화(FMT_MSA.3)
	TSF 데이터 관리(FMT_MTD.1)
	관리기능 명세(FMT_SMF.1)
	보안 역할(FMT_SMR.1)

TSF 데이터 훼손	인증(FIA_UAU.1)
	식별(FIA_UID.1)
	보안기능 관리(FMT_MOF.1)
	보안속성 관리(FMT_MSA.1)
	정적 속성 초기화(FMT_MSA.3)
	TSF 데이터 관리(FMT_MTD.1)
	관리기능 명세(FMT_SMF.1)
	보안 역할(FMT_SMR.1)

[표 1] 위협과 보안목적의 매핑

TOE의 보안기능이 사용하는 세부 보안기능은 [표 1]과 같이 도출된 보안기능 요구사항이 아래 [표 2]와 같이 추적됨을 알 수 있다.

보안기능	세부 보안기능	보안기능 요구사항
자체시험	체크포인트	FPT_FLS.1
	복구	FPT_RCV.2
	이벤트 모니터링	FPT_TST.1
향상된 스케줄러	우선순위 기반 스케줄링	FRU_PRS.1
	예외 상황 처리	FRU_RSA.1
식별 및 인증	식별자 및 비밀값 생성	FIA_UID.1
	식별자 중복 확인	
	인증	FIA_UAU.1
	인증 횟수 제한	
	인증 금지	
보안 속성 관리	커널의 관리기능 제어 및 명세	FMT_MOF.1
		FMT_MSA.1
		FMT_SMF.1
	커널의 보안 속성 제어	FMT_MSA.3
	커널의 보안 정적 속성 제어 및 Data 제어	FMT_MTD.1
	프로세스 행동 제어	FMT_SMR.1

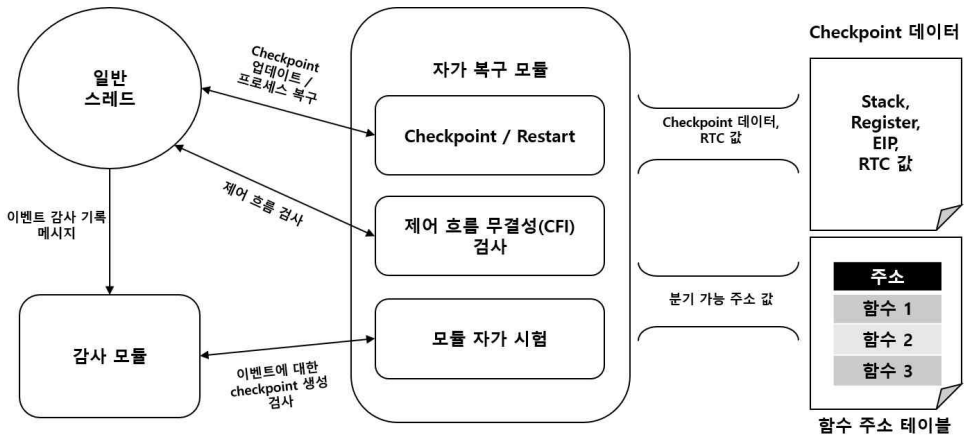
[표 2] 보안목적과 보안기능의 매핑

3. 보안 아키텍처 상세 설명

본 장에서는 2장에서 SFR과 매핑된 각 기능들이 해당 보안성을 어떻게 만족하는지 그 근거에 대해 설명한다. 3.1.에선 자체 보호, 3.2.에선 우회 불가성에 대해 설명한다.

3.1. 자체 보호

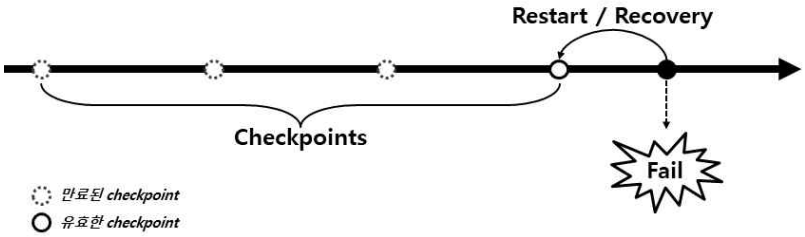
TOE에서는 현재 시스템 내 TSF 또는 일반 스레드가 불안전한 상태인지 안전한 상태인지 주기적으로 점검하여 시스템 내 TSF가 정상적인 서비스를 항상 제공할 수 있도록 보호한다. 이때 안전한 상태란 시스템의 흐름이 분기되는 시점에서 메모리상 안전한 주소로 분기되는 경우를 말하며, 메모리상 안전한 주소가 아닌 공격으로 인한 임의의 주소로 분기되거나 메모리 초기화 과정에서 설정된 Stack Guard 값이 임의의 값으로 변경될 경우 이를 불안전한 상태라고 말한다. 안전한 주소와 불안전한 주소의 경우 본 마이크로커널에 탑재될 TSF 제어 흐름에서 분기 가능한 주소를 컴파일 시점에서 미리 파악하여 분기 가능 주소 테이블을 정적으로 생성한다. 이후 분기가 발생할 때마다 분기 목적 주소와 분기 가능 주소 테이블을 비교하여 해당 분기가 올바른 분기인지 파악한다. 또한 Stack Guard의 경우 스레드가 새로 할당되는 시점에 스택 내에서 일반 변수가 저장되는 영역과 스레드의 제어흐름을 결정할 수 있는 정보(ret 주소 등)가 저장되는 영역 사이에 고유한 특정 값을 일정크기만큼 채워둔다. 이후 스택에 대한 접근이 발생할 때마다 해당 영역에 대한 검사를 수행하며 만약 해당 영역이 기존에 채워둔 값과 상이할 경우 허가되지 않은 침범이 일어났다고 간주하여 안전하지 않은 상태로 판단하게 된다. 일반 스레드가 아닌 TSF의 경우 TSF가 생성한 보안 이벤트 감사기록과 실제로 모니터링을 통해 얻은 감사기록을 서로 비교하여 일치할 경우 안전한 상태로 판단하게 되고, 그렇지 않을 경우 TSF가 올바르게 동작하지 못하여 불안전한 상태라고 판단하게 된다. 다음 [그림 3]은 자체보호 시스템 모델을 보여준다.



[그림 3] 자체보호 시스템 모델

자체 점검 수행결과 응용프로그램이 안전한 상태라고 판단될 경우 해당 시점을 가장 최근 복원지점(Checkpoint)로 지정하고, 해당 시점의 메모리 값들과 레지스터 값 등 해당 스레드의 제어 흐름에 관한 모든 정보를 백업한다. 이후 다음 자체 점검 시점에서 TSF

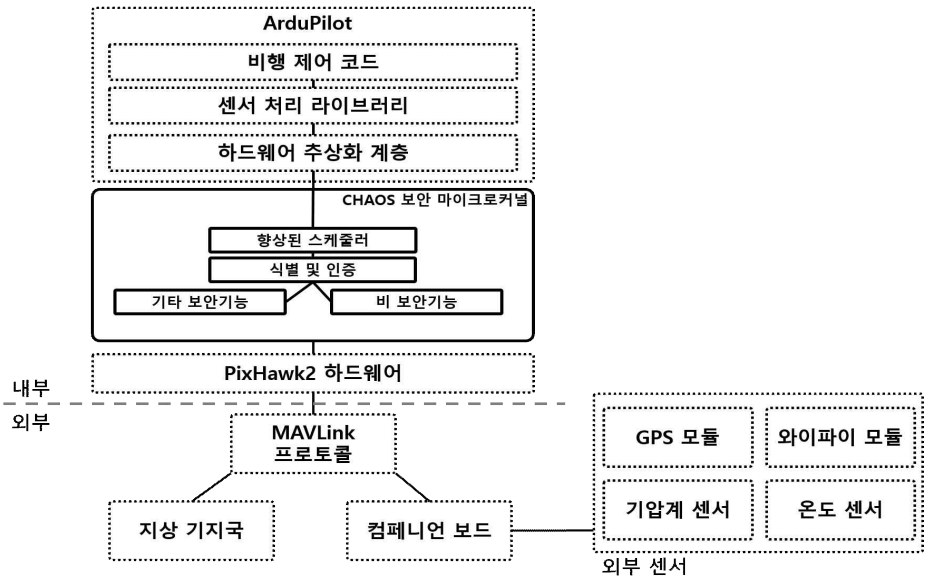
또는 응용프로그램이 불안전한 상태라고 판단될 경우 먼저 시스템은 무선 통신 스레드, 센서 처리 스레드와 같이 외부와 상호작용하는 스레드들을 Block 상태로 변환하여 안전모드로 진입한다. 이렇게 안전모드에 진입한 드론은 더 이상 외부로부터의 간섭을 받지 않도록 설정되어 안전하게 자가복구 기능을 수행할 수 있다. 안전모드에 진입한 후 불안전한 상태로 판단된 스레드의 종류에 따라 자가복구 과정이 두 가지로 나뉘게 된다. 첫 번째로, 불안전한 상태로 판단된 스레드가 자가복구를 수행하는 스레드 자신일 경우에는 사전에 프로그래머가 지정한 초기화 값을 이용하여 자기 자신을 초기화한다. 두 번째로, 타 스레드(예: 응용프로그램)가 불안전한 상태라고 판단된 경우 이전에 생성한 최근 복원지점으로 해당 스레드를 복원한다. 다음 [그림 4]는 자체보호 기능에서 사용하는 Checkpoint/Restart 메커니즘에 대해 간략하게 보여주고 있다.



[그림 4] Checkpoint/Restart 메커니즘

3.2. 우회 불가성

우회 불가성의 경우 드론 시스템에 대한 접근이 발생할 경우 TSF가 시스템에 대한 모든 접근을 관리하고 제어해야 한다는 것이다. 기존의 [그림 1]의 경우 드론 시스템의 구성을 논리적으로 표현하였기에 실제 데이터가 흐르는 순서가 올바르게 식별되지 않을 수 있기에 이를 물리적인 관점에서 시스템의 구성을 표현할 필요가 있다. 다음 [그림 5]는 [그림 1]의 전체 구성도를 실제 물리적 하드웨어 구성에 맞게 재배치 한 것이다.



[그림 5] 드론 시스템의 물리적 구성도

위 [그림 5]에 따르면 외부로부터 접근할 수 있는 접근 지점은 컴페니언 보드와 지상 기지국 두 가지가 있다. 이때 두 접근 지점은 하드웨어인 PixHawk의 Telemetry 포트를 통해서 통신을 하게되고 해당 Telemetry 포트로부터 입력되는 모든 데이터는 TOE의 향상된 스케줄러와 식별 및 인증기능을 제외한 기타 보안기능이나 비 보안기능을 거쳐 ArduPilot에 도달하기 때문에 TOE를 우회하여 처리될 수 있는 데이터는 존재하지 않는다. 따라서 이를 통해 TOE는 식별 및 인증 그리고 향상된 스케줄러 관련 TSF를 활용하여 내부로 유입되는 모든 데이터를 점검할 수 있기에 우회 불가성을 입증할 수 있다.