소프트웨어 코드 안전성분석 보고서

수정일자	수정자	버전	추가/수정 항목	내 용
2020-11-24	최홍준	v1.0		초안 작성
2020-11-25	이종훈	v2.0	SW 코드 안전성분석에 사용된 입력자료	
2020-11-25	최홍준	v3.0	코드 안전성 분석기법	

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

목차

표 목차

그림 목차

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

I. 목적

본 보고서는 고등급 보안 마이크로 커널(CHAOS)의 결함을 개발 초기에 진단하고 방지하기 위해 기능안전성 공통 표준인 IEC 61508에 근거하여 수행한 뮤텍스 코드 안전성분석 활동 내용을 기술하고, 그 결과를 분석하는 것에 목적이 있다.

II. 범위

본 보고서는 고등급 보안 마이크로 커널(CHAOS) 개발을 위한 뮤텍스 코드 안전성분석 수행의 범위를 다음과 같이 정의한다.

chmtx.h(뮤텍스 매크로 및 구조)

chschd.h(스케줄러 매크로 및 구조)

chsys.h(시스템 관련 매크로 및 구조)

chmtx.c(뮤텍스 코드)

III. 용어 정의 및 약어

소프트웨어 코드 안전성분석 보고서에 사용된 용어 및 약어를 정의하고 설명한다.

1. Safety function(안전 기능) : 특정 위험한 사건에 대하여 E/E/PE 시스템 또는 기타 위험성 감소 조치를 위해 구현된 기능
2. Fault(결함) : 요청된 기능을 수행할 때 기능 유닛의 능력이 감소하거나, 손실이 발생하는 비정상적 조건
3. Bug(버그) : 프로그램이 제대로 작동하지 않거나 잘못된 결과 또는 충돌을 일으키는 오류, 결함
4. Weakness(약점) : 소프트웨어 또는 하드웨어의 구현, 코드, 설계 또는 아키텍처의 결점, 결함, 버그, 취약점 또는 기타 오류
5. Static Analysis(정적분석) : 프로그램 실행 없이 프로그램 코드만을 가지고 문법, 코딩 규칙, 실행 오류 등 약점을 자동으로 식별하는 방법

6. Overflow(오버플로우) : 소프트웨어가 처리할 수 있는 범위를 벗어난 숫자, 값 또는 변수를 받을 때 발생하고 비정상적인 값을 반환

7. Division by zero(0으로 나누기) : 어떤 숫자를 0으로 나누는 나눗셈을 의미하며, 양수 또는 음수 무한대를 생성하여 예외와 에러 메시지를 생성

8. Out of bounds array access(범위에 벗어난 배열 접근) : 소프트웨어는 인덱스를 수정하거나 버퍼 경계 외부에 있는 메모리 위치를 참조할 때, 정의되지 않았거나 예기치 않은 결과를 생성

IV. 시스템 개요

본 보고서에서 안전성 분석의 대상인 마이크로 커널은 드론을 제어하기 위해 탑재되는 소프트웨어이며 상위수준 시스템 구조는 그림 1과 같다.

그림 1. 상위수준 시스템 구조

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

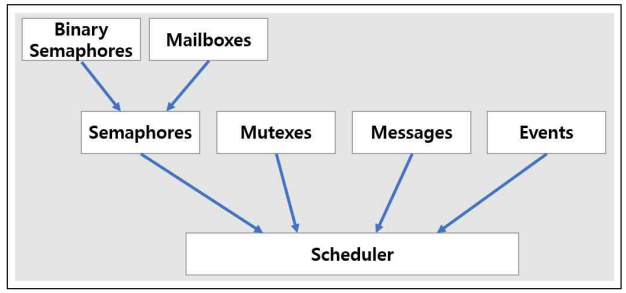

그림 2. CHAOS 커널레벨 시스템 구조

커널 레벨 시스템에는 그림 2와 같은 모듈로 구성되어 있다. 세마포어, 뮤텍스, 메시지, 이벤트 등의 모듈은 스케줄러에 의해 동작하며 이진 세마포어와 메일박스는 세마포어에 의해 동작한다.

스케줄러 시스템은 그림 3과 같이 구성되어 있으며 모든 요소에 대한 시간제한 기능을 구현하기 위해 가상 타이머와 긴밀하게 결합 되어있다.

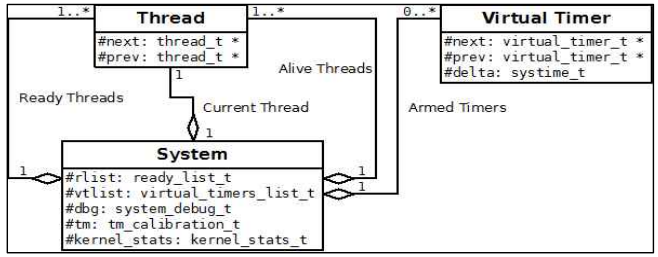

그림 3. CHAOS 스케줄러 시스템 구조

각각의 스레드는 Current, Alive, Ready 상태로 존재하는 스레드로 구분되며 Ready 상태의 스레드는 그림 4와 같이 우선순위에 의해 정렬되어 있다. 시스템은 항상 “유휴 스레드”라는 특수 스레드를 실행하며 유휴 스레드를 실행함으로써 시스템 전력 소비를 줄일 수 있으며 유휴 스레드는 오직 Ready, Current 상태에만 있을 수 있고 Sleep, Terminate 상태로 갈 수 없다.

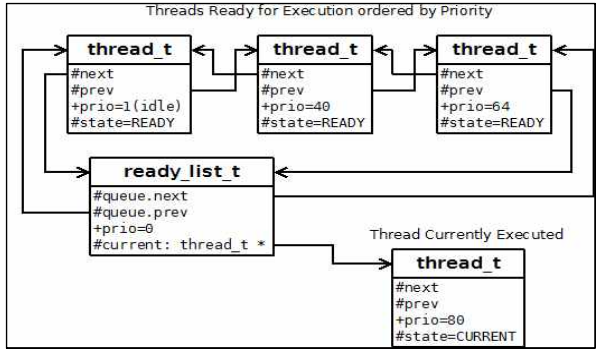

그림 4. CHAOS 스레드 구조

시스템 구조는 아래 [그림 4]와 같다. 드론 하드웨어는 Pixhawk2를 사용한다. 이 장비는 다중 GPS 연결 포트 이외에도 기타 센서 및 장비와 유선 및 무선 통신을 할 수 있도록 Carrier Board를 통해 IC2(Inter Integrated Circuit), CAN(Controller Area Network) 포트와 MAVLink와 같은 무선 통신 프로토콜을 지원하기 위한 Telemetry 포트를 지원한다. 안정성 분석을 수행하는 대상인 Drone OS가 마이크로커널이 탑재되는 영역이다. 실제 비행 제어를 수행하는 시스템은 Flight Computer로 Flight Controller, Drone HW를 담당한다. 그리고 다양한 Application과 Companion Computer OS, SW, HW를 담당하는 영역인 Companion Computer가 있다. 드론 하드웨어는 MAVLink를 사용하여 Companion Computer와 Ground Control Station(GCS)와 통신한다.

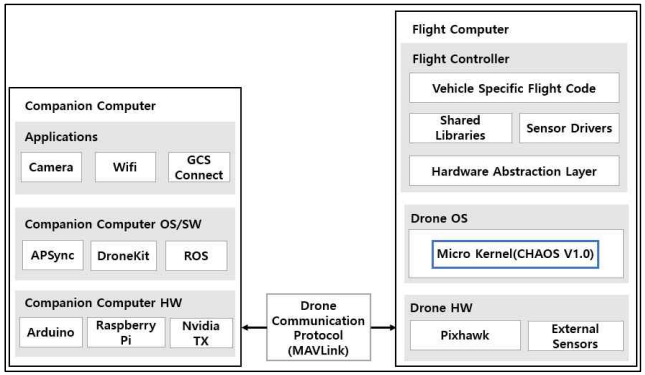

그림 5. 소프트웨어 구조설계

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

V. 코드 안전성 분석기법

가. SW 코드 안전성분석에 사용된 입력자료

고등급 보안 마이크로 커널(CHAOS)의 뮤텍스 코드 안전성분석을 위해 사용된 입력자료를 설명한다.

① 안전기능 요구사항 : 뮤텍스와 관련된 우선순위 반전 문제 해결을 위한 안전기능 요구사항

② 시스템 상태도 : 전체적인 시스템 동작 과정과 임계영역을 파악하기 위한 시스템 상태도

③ 모듈별 함수 호출 그래프 및 순서도 : 모듈별 함수들의 입/출력과 기능, 관련 커널 오브젝트를 파악하기 위한 자료

④ 데이터 구조도 : 뮤텍스 관련 변수 및 구조체의 데이터 구조도

⑤ 고등급 보안 마이크로 커널(CHAOS)의 뮤텍스 관련 코드

나. SW 코드 안전성분석에 사용된 기법

SW 코드 안전성분석에 사용된 기법은 정적분석이다. 정적분석은 프로그램이 출시되기 전에 정적분석 도구를 활용하여 문법, 코딩규칙, 실행 오류 등 약점을 자동으로 식별하는 방법이다. 정적분석 도구는 프로그램 실행 없이 프로그램 코드만을 가지고 정적으로 검사한다. 도구별 체커와 규칙에 따라, 정적분석을 수행하기에 모든 약점을 진단할 수 있는 것은 아니다. 또한, 정적분석 도구에서 알람이 발생했다고 해서 모든 알람이 정탐은 아니다. 이 같은 단점이 존재하지만, 정적분석을 통해 비교적 개발 초기에 결함을 발견하고 개발 효율성과 비용 절감의 장점이 존재하여 정적분석 도구가 많이 활용된다. 또한, 동적분석을 통해 발견하기 힘든 오버플로우, 0으로 나누기, 범위에 벗어난 배열 접근 등의 약점을 효과적으로 진단할 수 있다.

다. SW 코드 안전성분석에 사용된 절차

SW 코드안전성 분석은 크게 진단준비, 진단계획, 진단수행 및 진단종료 절차로 진행하였다.

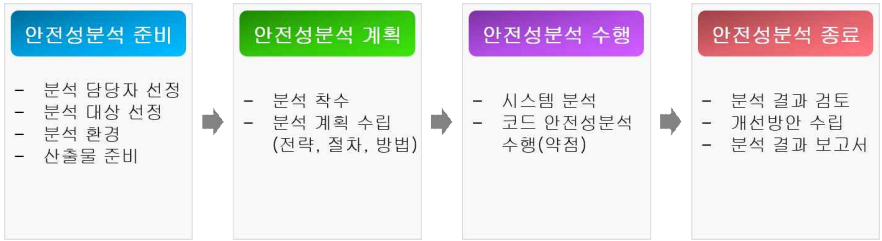

그림 6. SW 코드 안전성분석 절차

1) SW 코드 안전성분석 준비

SW 코드 안전성분석 준비 단계에서 먼저 분석 담당자를 선정하고 분석 대상을 선정하였다. 분석 대상은 ChibiOS의 뮤텍스 관련 코드와 보안 요구사항과 안전기능이 추가된 CHAOS 뮤텍스 관련 코드이다. 또한, 이때까지 나온 산출물을 기반으로 대상 시스템을 파악한다.

표 1. SW 코드 안전성분석 환경

Option	Value
-Ubuntu	gnu7.x
-compiler	gnu7.x
-context-sensitivity-auto	true
-dos	true
-float-rounding-mode	to-nearest
-lang	C
-main-generator	true
-main-generator-calls	unused
-signed-integer-overflows	forbid
-target	i386
-to	Software Safety Analysis level 2
-uncalled-function-checks	none
-unsigned-integer-overflows	allow
-verif-version	1.0

표 1은 Polyspace 설정 내용이다. 분석 대상의 언어는 C이고 gnu7.x 컴파일러 사용하여 빌드했다. 또한, 검증 레벨은 Software Safety Anlysis level 2를 사용하였다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

2) SW 코드 안전성분석 계획

분석 담당자, 시스템 이해 관계자 등 분석 이해관계자 간 사전 회의를 통해 분석을 시작하고 SW 코드 안전성분석에 사용된 입력자료를 통해 대상 시스템에 대해 예비 분석 및 분석 전략 등 세부 추진 계획를 수립하였다.

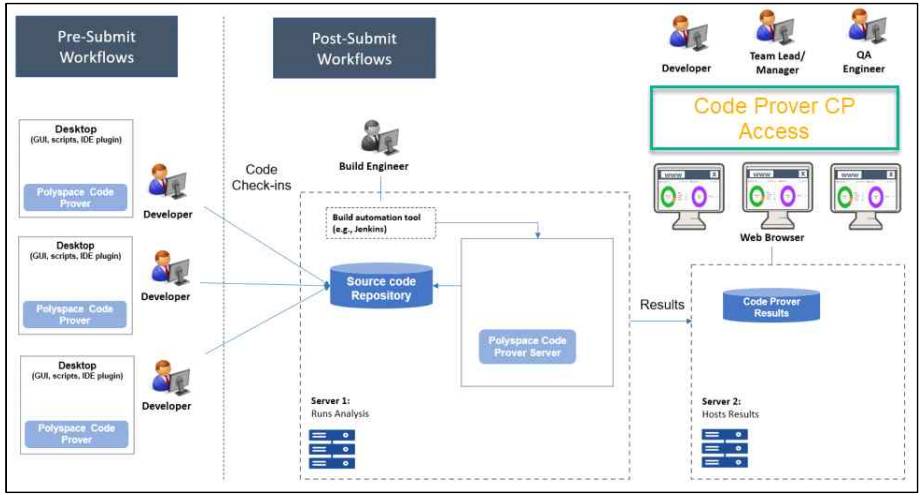

그림 7. SW 코드 안전성분석 프로세스

그림 7는 SW 코드 안전성분석 프로세스로 개발자는 소스 코드 저장소에 CHAOS를 업로드하고 코드 안전성분석 담당자는 Polyspace Code Prover 도구를 이용하여 저장소에 있는 코드를 빌드하고 정적분석한다. 또한, 코드 검증 결과 보고서를 깃허브에 공유하여 웹페이지 html 형태로 볼 수 있다.

3) SW 코드 안전성분석 수행

먼저 산출물 및 시스템 상태도, 모듈별 함수 호출 그래프 및 순서도 등과 입력자료를 통해 시스템에 대해 상세 분석하였다. 분석한 내용을 바탕으로 코드 안전성분석을 수행하였다.

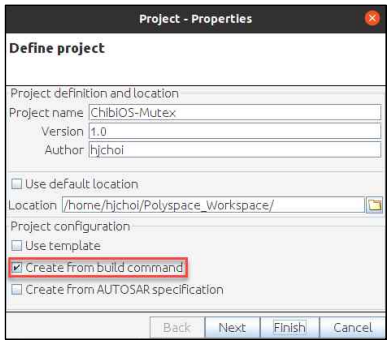

그림 8. 분석 대상 프로젝트 생성

분석 대상을 정적분석하기 위해 Polyspace 도구에서 프로젝트를 그림 8과 같이 생성한다. 임베디드 소프트웨어의 경우 빌드 과정이 반드시 필요하기에 'Create from build command' 버튼을 클릭하여 프로젝트를 만든다.

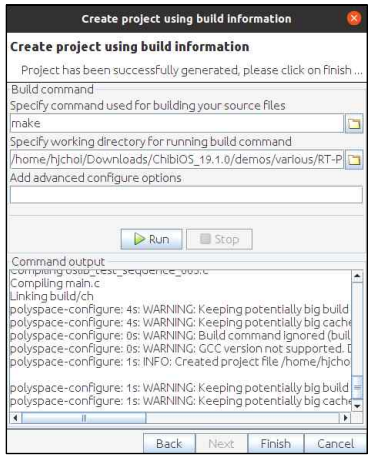

그림 9. 프로젝트 빌드

프로젝트 빌드 하는데 사용하는 명령어는 make를 사용한다. 빌드하는 폴더는 하드웨어 사양에 맞추어 선택하면 되는데, RT-Posix-Simulator를 선정하였다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

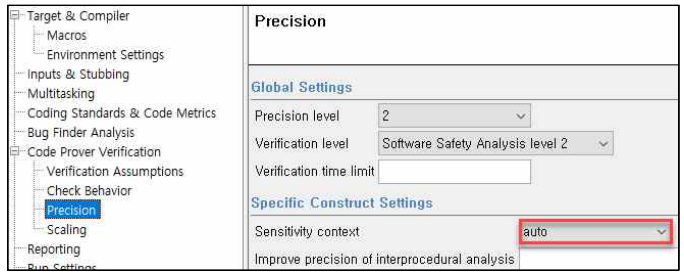

그림 10. Sensitivity context 설정

Polyspace Code Prover 도구는 정적분석 시 자동으로 main 함수를 생성한다. 이 main 함수는 대상 소스 코드의 함수를 호출하게 되어, 경로에 따라 결과가 달라질 수 있다. 그림 10과 같이 'Sensitivity context'를 auto로 설정하면 도구에서 알람이 발생하였을 때, 어떤 경로로 함수가 호출되었는지 확인할 수 있다.

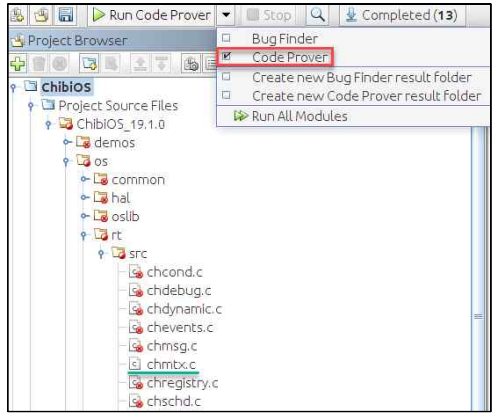

그림 11. 파일 선택 및 Code Prover 실행

빌드하면 폴더 내의 모든 파일이 업로드되고 정적분석 시 대상 파일만 포함할 수 있다.

4) SW 코드 안전성분석 종료

SW 코드 안전성분석을 종료하고 결과 보고서를 받아 분석 이해관계자가 볼 수 있게 깃허브에 공유하였다. 분석 결과 및 개선방안의 적절성은 추후 검토 예정이다.

라. SW 코드 안전성분석에 사용된 도구

SW 코드 안전성분석에 사용된 도구는 MathWorks 사의 Polyspace Code Prover이다. Polyspace Code Prover는 정형기법 기반의 추상적 해석(Abstract Interpretation)을 사용하여 발생할 수 있는 모든 제어 흐름과 데이터 흐름을 분석하여 C, C++ 소스코드에서 오버플로우, 0으로 나누기, 범위에 벗어난 배열 접근 등 중대한 런타임 에러의 존재를 증명한다. 그리고 29개의 체커를 보유하고 있으며, 미탐(False Negatives)이 존재하지 않는다. 그림 4는 Polyspace Code Prover의 개요이다.

Feature	Bug Finder	Code Prover
Number of checkers	288	29 (Critical subset)
Depth of analysis	Fast. For instance: • Faster analysis. • Easier set up and review. • Fewer runs for clean code. • Results in real time.	Exhaustive. For instance: • All operations of a type checked for certain critical errors. • More rigorous data and control flow analysis.
Reporting criteria	Probable defects	Proven findings
Bug finding criteria	Few false positives	Zero false negatives

그림 12. Polyspace Code Prover 개요

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

VI. 코드 안전성 분석결과

Polyspace Code Prover를 사용해 스케줄러 우선순위 반전문제 코드가 있는 chmtx.c 파일을 보안 요구사항과 안전기능 포함 전후로 나누어 정적분석 수행하고 결과를 정리한다.

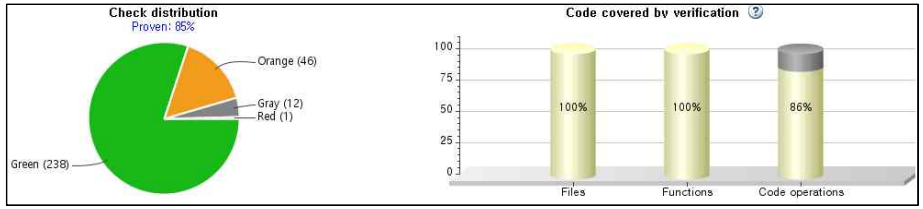

그림 13. ChibiOS chmtx.c 정적분석 결과 개요

Polyspace Code Prover를 사용해 스케줄러 우선순위 반전문제 코드가 있는 chmtx.c 파일을 보안 요구사항과 안전기능 포함 전후로 나누어 정적분석 수행하고 결과를 정리한다.

File	Proven	Green	Red	Gray	Orange
chmtx.h	100.0%	2	0	0	0
__polyspace_main.c	100.0%	1	0	0	0
chsys.h	100.0%	0	0	1	0
chmtx.c	86.5%	206	0	11	34
chschd.h	71.4%	29	1	0	12
Total	**84.5%**	**238**	**1**	12	46

그림 14. ChibiOS chmtx.c 관련 파일별 정적분석 결과

그림 14는 ChibiOS chmtx.c 관련 파일별 정적분석 결과이다. chmtx.c 파일은 Orange가 34개로 Illegally dereferenced pointer 14개, Non-initialized pointer 14개, Non-initialized variable 6개의 약점이 존재한다. chschd.h 파일은 Red에서 Illegally dereferenced pointer 1개, Orange에서 Illegally dereferenced pointer 5개, Non-initialized pointer 7개로 총 12개의 약점이 존재한다.

표 2. ChibiOS chschd.h 증명된 런타임 에러

Check	Function	Line	Detail
Illegally dereferenced pointer	queue_prio_insert()	590	Error: pointer is outside its bounds

표 2는 그림 14의 Red 알람이다. 증명된 런타임 에러로 Illegally dereferenced pointer 체커에 의해 ChibiOS chschd.h 파일의 queue_prio_insert 함수 590번 라인에서 약점이 발생하였다.

표 3. ChibiOS chsys.h 증명된 도달하지 않는 코드

Check	Function	Line	Detail
Unreachable code	chSysUnlock()	374	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 374 (column 11). Block ends at line 374 (column 184)

표 3은 그림 14의 Gray 알람 일부이다. Unreachable code 체커에 의해 ChibiOS chschd.h 파일의 queue_prio_insert 함수 374번 라인에서 약점이 발생하였다.

표 4. ChibiOS chmtx.c 증명된 도달하지 않는 코드

Check	Function	Line	Detail
Unreachable code	chMtxObjectInit()	105	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 105 (column 11). Block ends at line 105 (column 72)
Unreachable code	chMtxLockS()	143	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 143 (column 11). Block ends at line 143 (column 72)
Unreachable code	chMtxTryLockS()	285	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 285 (column 11). Block ends at line 285 (column 72)
Unreachable code	chMtxUnlock()	327	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 327 (column 11). Block ends at line 327 (column 72)
Unreachable code	chMtxUnlock()	331	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 331 (column 11). Block ends at line 331 (column 82)

Unreachable code	chMtxUnlock()	332	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 332 (column 11). Block ends at line 332 (column 81)
Unreachable code	chMtxUnlock()	339	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 339 (column 13). Block ends at line 339 (column 75)
Unreachable code	chMtxUnlockS()	415	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 415 (column 11). Block ends at line 415 (column 72)
Unreachable code	chMtxUnlockS()	417	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 417 (column 11). Block ends at line 417 (column 82)
Unreachable code	chMtxUnlockS()	418	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 418 (column 11). Block ends at line 418 (column 81)
Unreachable code	chMtxUnlockS()	425	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 425 (column 13). Block ends at line 425 (column 75)

표 4는 그림 14의 Gray 알람 일부이다. ChibiOS chmtx.c 파일의 Unreachable code이다.

표 5. ChibiOS chschd.h 증명되지 않은 런타임 에러

Check	Function	Line	Detail
Illegally dereferenced pointer	queue_notempty()	566	Warning: pointer may be outside its bounds
Non-initialized pointer	queue_notempty()	566	Warning: pointer may be non-initialized
Non-initialized pointer	queue_fifo_remove()	607	Warning: pointer may be non-initialized
Illegally dereferenced pointer	queue_fifo_remove()	609	Warning: pointer may be outside its bounds
Non-initialized pointer	queue_fifo_remove()	609	Warning: pointer may be non-initialized
Illegally dereferenced pointer	queue_fifo_remove()	610	Warning: pointer may be outside its bounds
Non-initialized pointer	queue_dequeue()	626	Warning: pointer may be non-initialized
Illegally dereferenced pointer	queue_dequeue()	626	Warning: pointer may be outside its bounds
Non-initialized pointer	queue_dequeue()	626	Warning: pointer may be non-initialized
Non-initialized pointer	queue_dequeue()	627	Warning: pointer may be non-initialized
Illegally dereferenced pointer	queue_dequeue()	627	Warning: pointer may be outside its bounds
Non-initialized pointer	queue_dequeue()	627	Warning: pointer may be non-initialized

표 5는 그림 14의 Orange 알람 일부이다. ChibiOS chschd.h 파일의 Illegally dereferenced pointer, Non-initialized pointer 약점이다.

표 6. ChibiOS chmtx.c 증명되지 않은 런타임 에러

Check	Function	Line	Detail
Illegally dereferenced pointer	chMtxLockS()	165	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxLockS()	165	Warning: variable may be non-initialized (type: unsigned int 32)
Illegally dereferenced pointer	chMtxLockS()	165	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxLockS()	170	Warning: variable may be non-initialized (type: unsigned int 8)

Non-initialized pointer	chMtxLockS()	173	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxLockS()	193	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxLockS()	233	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxTryLockS()	306	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlock()	344	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlock()	359	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlock()	359	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlock()	359	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized pointer	chMtxUnlock()	360	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlock()	360	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlock()	360	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized pointer	chMtxUnlock()	362	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxUnlock()	376	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockS()	430	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockS()	445	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockS()	445	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlockS()	445	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized pointer	chMtxUnlockS()	446	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockS()	446	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlockS()	446	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized	chMtxUnlockS()	448	Warning: pointer may be

pointer			non-initialized
Non-initialized pointer	chMtxUnlockS()	462	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockAllS()	490	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockAllS()	492	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockAllS()	492	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxUnlockAllS()	499	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockAll()	528	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockAll()	531	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockAll()	531	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxUnlockAll()	538	Warning: pointer may be non-initialized

표 6은 그림 14의 Orange 알람 일부이다. ChibiOS chmtx.c 파일의 Illegally dereferenced pointer, Non-initialized pointer, Non-initialized variable 약점이다.

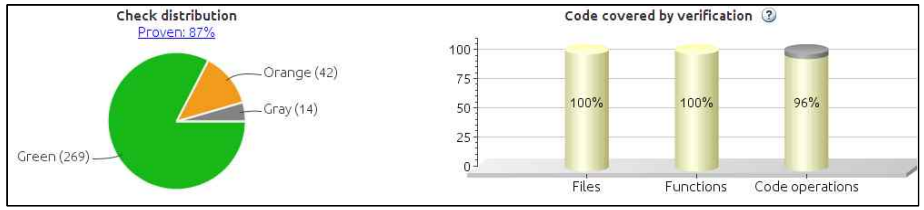

그림 15. CHAOS chmtx.c 정적분석 결과 개요

그림 15은 보안 요구사항과 안전기능이 포함된 chmtx.c 파일을 정적분석한 결과이다. Code prover는 Orange에서 Illegally dereferenced pointer 24개 존재, Non-initialized pointer 12개, Non-initialized variable 6개 총 42개의 약점을 발견했다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈, 최홍준	소속	고려대학교	연구 책임자	김승주
	작성일	2020-11-23	파일명	2020-기술문서04-SW코드안전성분석보고서.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

File	Proven	Green	Red	Gray	Orange
chschd.h	100.0%	59	0	0	0
chmtx.h	100.0%	2	0	0	0
_polyspace_main.c	100.0%	1	0	0	0
chsys.h	100.0%	0	0	1	0
chmtx.c	84.0%	207	0	13	42
Total	**87.1%**	**269**	**0**	**14**	**42**

그림 16. CHAOS chmtx.c 관련 파일별 정적분석 결과

그림 16는 ChibiOS chmtx.c에서 발견된 Red 알람이 제거됨을 보여준다.

표 7. CHAOS chsys.h 증명된 도달하지 않는 코드

Check	Function	Line	Detail
Unreachable code	chSysUnlock()	374	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 374 (column 11). Block ends at line 374 (column 184)

표 7은 그림 16의 Gray 알람 일부이다. CHAOS chsys.h 파일의 Unreachable code이다.

표 8. CHAOS chmtx.c 증명된 도달하지 않는 코드

Check	Function	Line	Detail
Unreachable code	chMtxObjectInit()	105	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 105 (column 11). Block ends at line 105 (column 72)
Unreachable code	chMtxLockS()	143	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 143 (column 11). Block ends at line 143 (column 72)
Unreachable code	chMtxLockS()	218	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 218 (column 15). Block ends at line 218 (column 75)
Unreachable code	chMtxLockS()	219	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false

			at line 219 (column 15). Block ends at line 219 (column 77)
Unreachable code	chMtxTryLockS()	285	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 285 (column 11). Block ends at line 285 (column 72)
Unreachable code	chMtxUnlock()	327	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 327 (column 11). Block ends at line 327 (column 72)
Unreachable code	chMtxUnlock()	331	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 331 (column 11). Block ends at line 331 (column 82)
Unreachable code	chMtxUnlock()	332	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 332 (column 11). Block ends at line 332 (column 81)
Unreachable code	chMtxUnlock()	339	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 339 (column 13). Block ends at line 339 (column 75)
Unreachable code	chMtxUnlockS()	415	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 415 (column 11). Block ends at line 415 (column 72)
Unreachable code	chMtxUnlockS()	417	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 417 (column 11). Block ends at line 417 (column 82)
Unreachable code	chMtxUnlockS()	418	The section of code is unreachable or

			the condition is redundant. If-condition always evaluates to false at line 418 (column 11). Block ends at line 418 (column 81)
Unreachable code	chMtxUnlockS()	425	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 425 (column 13). Block ends at line 425 (column 75)

표 8은 그림 16의 Gray 알람 일부이다. CHAOS chmtx.c 파일의 Unreachable code이다.

표 9. CHAOS chmtx.c 증명되지 않은 런타임 에러

Check	Function	Line	Detail
Illegally dereferenced pointer	chMtxLockS()	165	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxLockS()	165	Warning: variable may be non-initialized (type: unsigned int 32)
Illegally dereferenced pointer	chMtxLockS()	165	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxLockS()	170	Warning: variable may be non-initialized (type: unsigned int 8)
Non-initialized pointer	chMtxLockS()	173	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxLockS()	174	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxLockS()	174	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxLockS()	174	Warning: pointer may be non-initialized
Non-initialized pointer	chMtxLockS()	193	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxLockS()	233	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxTryLockS()	306	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxTryLockS()	307	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlock()	344	Warning: pointer may be outside its bounds

Illegally dereferenced pointer	chMtxUnlock()	359	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlock()	359	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlock()	359	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlock()	359	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized pointer	chMtxUnlock()	360	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlock()	360	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlock()	360	Warning: variable may be non-initialized (type: unsigned int 32)
Illegally dereferenced pointer	chMtxUnlock()	362	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlock()	362	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlock()	376	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockS()	430	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockS()	445	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockS()	445	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockS()	445	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlockS()	445	Warning: variable may be non-initialized (type: unsigned int 32)
Non-initialized pointer	chMtxUnlockS()	446	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockS()	446	Warning: pointer may be outside its bounds
Non-initialized variable	chMtxUnlockS()	446	Warning: variable may be non-initialized (type: unsigned int 32)
Illegally dereferenced pointer	chMtxUnlockS()	448	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockS()	448	Warning: pointer may be non-initialized
Illegally	chMtxUnlockS()	462	Warning: pointer may be outside its

dereferenced pointer			bounds
Illegally dereferenced pointer	chMtxUnlockAllS()	490	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockAllS()	492	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockAllS()	492	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockAllS()	499	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockAll()	528	Warning: pointer may be outside its bounds
Illegally dereferenced pointer	chMtxUnlockAll()	531	Warning: pointer may be outside its bounds
Non-initialized pointer	chMtxUnlockAll()	531	Warning: pointer may be non-initialized
Illegally dereferenced pointer	chMtxUnlockAll()	538	Warning: pointer may be outside its bounds

표 9는 그림 16의 Orange 알람이다. CHAOS chmtx.c 파일의 Illegally dereferenced pointer, Non-initialized pointer, Non-initialized variable 약점이다.

참고문헌

[1] IEC, IEC61508. “61508 functional safety of electrical/electronic/programmable electronic safety-related systems.” International electrotechnical commission (1998).

[2] ISO, ISO26262. “26262: Road vehicles-Functional safety.” International Standard ISO/FDIS 26262 (2011).

[3] SW 안전성 공통 개발 가이드. “정보통신산업진흥원.” (2016).

[4] 소프트웨어 안전 진단 가이드. “한국정보통신기술협회.” (2016).

[5] 전자정부 SW 개발·운영자를 위한 소프트웨어 개발보안 가이드. “한국인터넷진흥원.” (2019).

[6] Polyspace Code Prover Getting Started Guide. “MathWorks.” (2020).

[7] MITRE. Common Weakness Enumeration: CWE. https://cwe.mitre.org/ (2020).

[8] Ayewah, N., Pugh, W., Hovemeyer, D., Morgenthaler, J. D., & Penix, J. Using static analysis to find bugs. IEEE software, 25(5), 22-29. (2008).

[9] 박정현, 박영식, 정효택. SW 개발 R&D 프로젝트에서 소스 코드 품질을 위한 정적 분석. 전자통신동향분석, 32(1), 102-115. doi:10.22648/ETRI.2017.J.320111. (2017).

[10] ChibiOS free embedded RTOS - ChibiOS Homepage. [online] Available at: https://www.chibios.org/dokuwiki/doku.php [Accessed 25 Nov. 2020].

