

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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	
2021-11-12	윤성호	v3.1	CHAOS v0.1 정적 분석	
2021-12-15	윤성호	v4.0	CHAOS v0.2 정적 분석	

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보 고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

목차

I. 목적	3
II. 범위	3
III. 용어 정의 및 약어	4
IV. 시스템 개요	4
V. 코드 안전성 분석기법	7
가. SW 코드 안전성분석에 사용된 입력자료	7
나. SW 코드 안전성분석에 사용된 기법	7
다. SW 코드 안전성분석에 사용된 절차	9
1) SW 코드 안전성분석 준비	9
2) SW 코드 안전성분석 계획	10
3) SW 코드 안전성분석 수행	10
4) SW 코드 안전성분석 종료	13
라. SW 코드 안전성분석에 사용된 도구	13
VI. 코드 안전성 분석결과	14
가. ChibiOS CodeProver 수행결과	14
나. CHAOS version 0.1 CodeProver 수행결과	26
다. CHAOS version 0.1 BugFinder 수행결과	34
라. CHAOS version 0.2 CodeProver 수행결과	36
마. CHAOS version 0.2 BugFinder 수행결과	44
<부록 1> CHAOS version 0.1 CodeProver 결과 분석	46
<부록 2> CHAOS version 0.1 BugFinder 결과 분석	88
<부록 3> CHAOS version 0.2 CodeProver 결과 분석	90
<부록 4> CHAOS version 0.2 BugFinder 결과 분석	138

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보 고서-v3.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 안전성분석 활동 내용을 기술하고, 그 결과를 분석하는 것에 목적이 있다.

II. 범위

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

chaudit.c

chcond.c

chdebug.c

chdynamic.c

chevetns.c

chfia.c

chmsg.c

chmtx.c

chregistry.c

chschd.c

chsecure.c

chstats.c

chsys.c

chthreads.c

chtm.c

chtrace.c

chvt.c

III. 용어 정의 및 약어

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

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	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

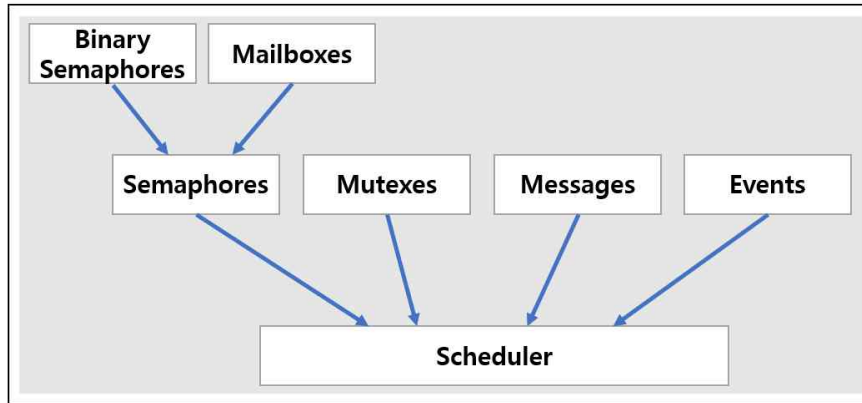


그림 2. CHAOS 커널레벨 시스템 구조

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.

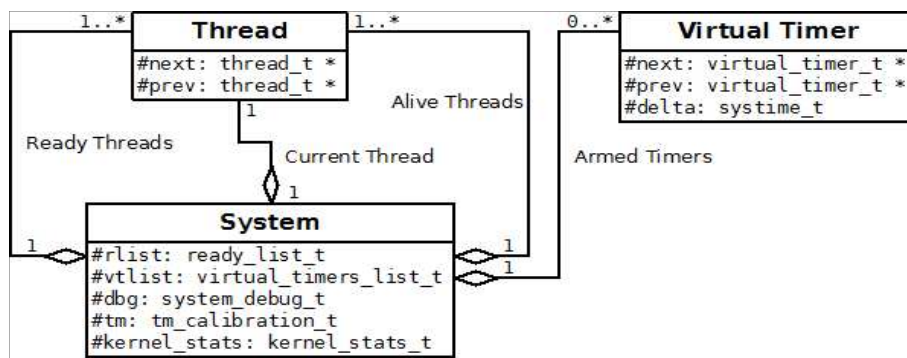


그림 3. CHAOS 스케줄러 시스템 구조

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.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

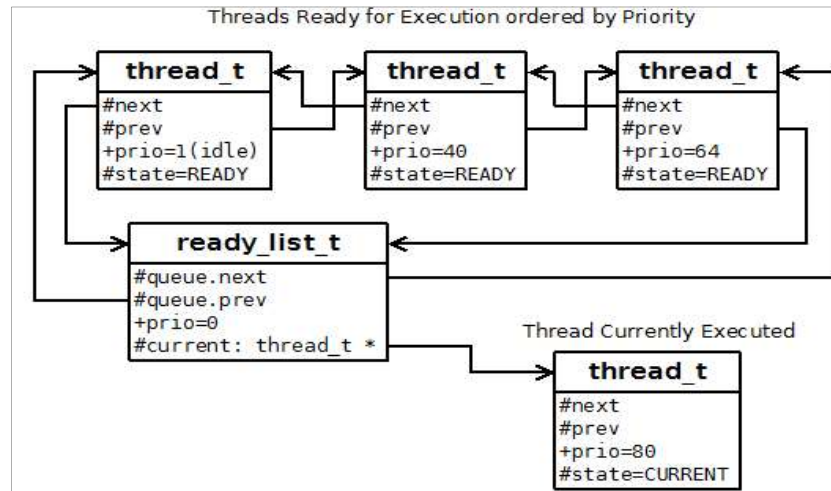


그림 4. CHAOS 스레드 구조

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.

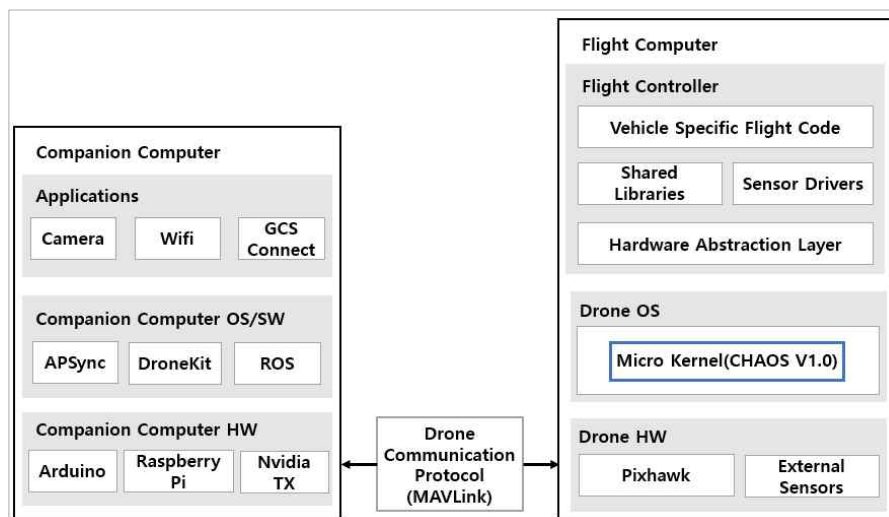


그림 5. 소프트웨어 구조설계

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.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.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

다. SW 코드 안전성분석에 사용된 절차

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

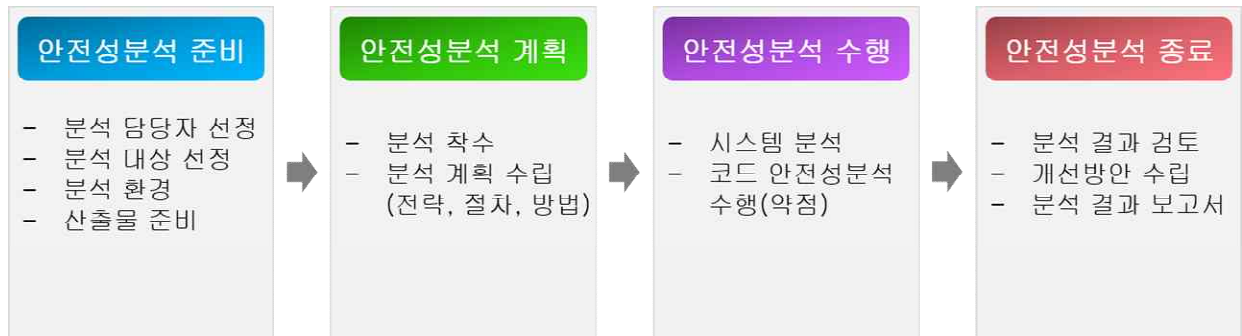


그림 6. SW 코드 안전성분석 절차

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	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.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 계획을 수립하였다.

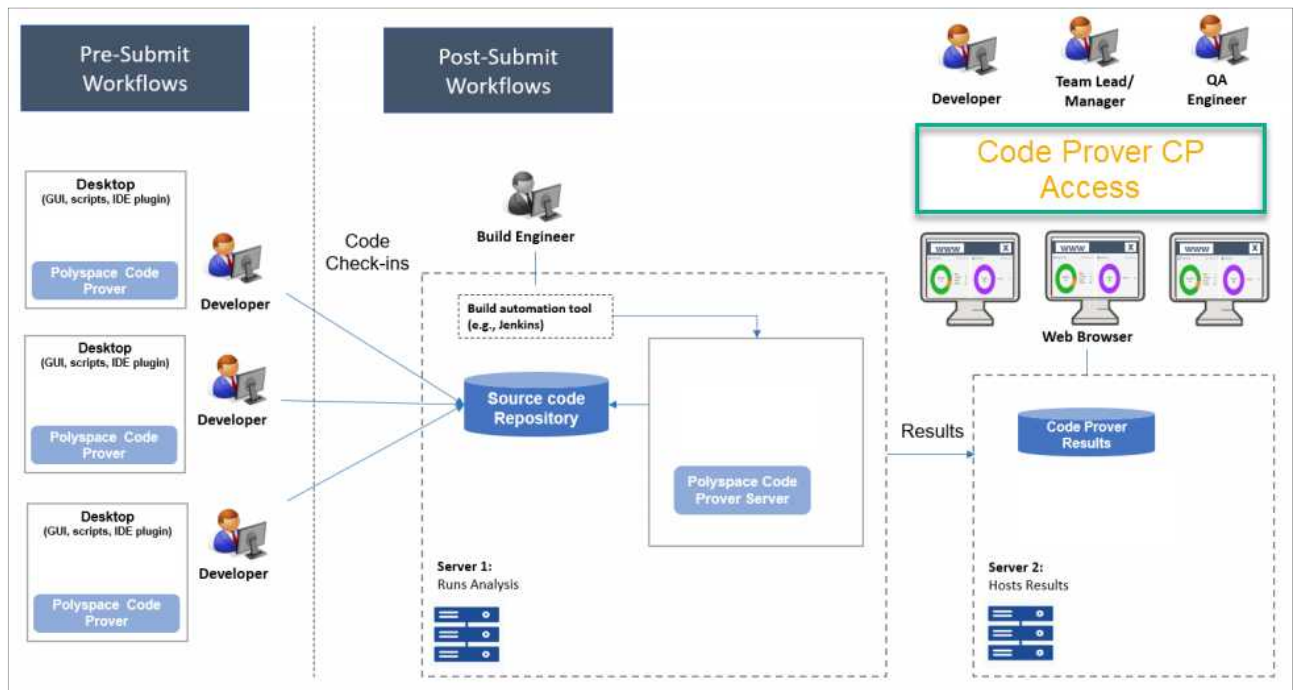


그림 7. SW 코드 안전성분석 프로세스

그림 7는 SW 코드 안전성분석 프로세스로 개발자는 소스 코드 저장소에 CHAOS를 업로드 하고 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.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

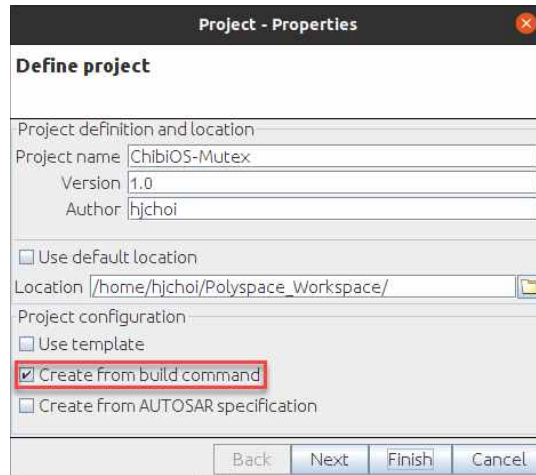


그림 8. 분석 대상 프로젝트 생성

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.

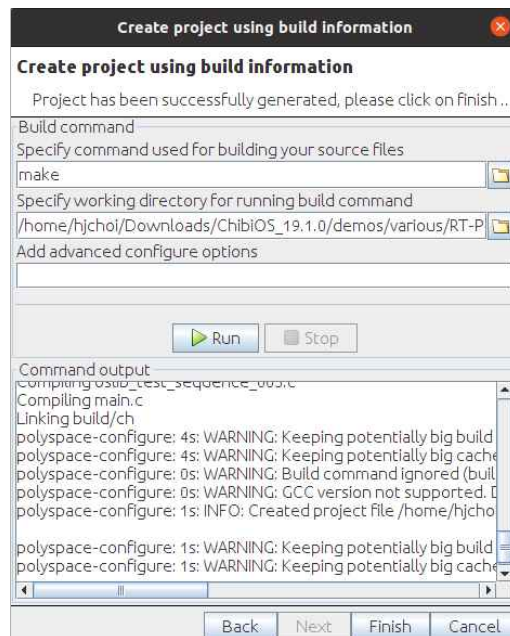


그림 9. 프로젝트 빌드

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

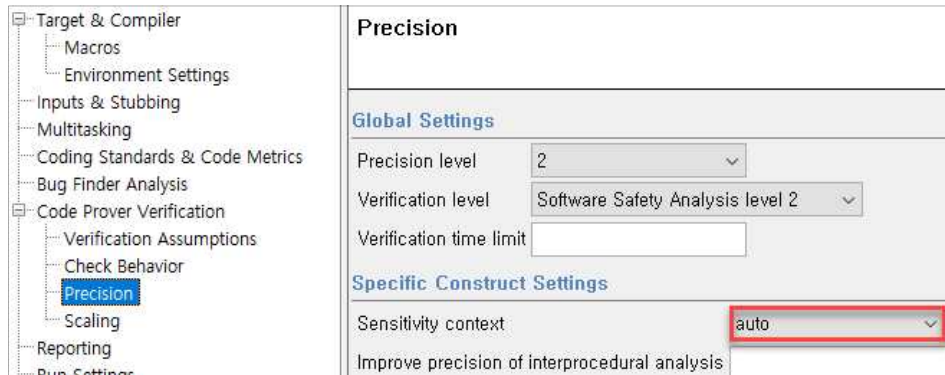


그림 10. Sensitivity context 설정

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.

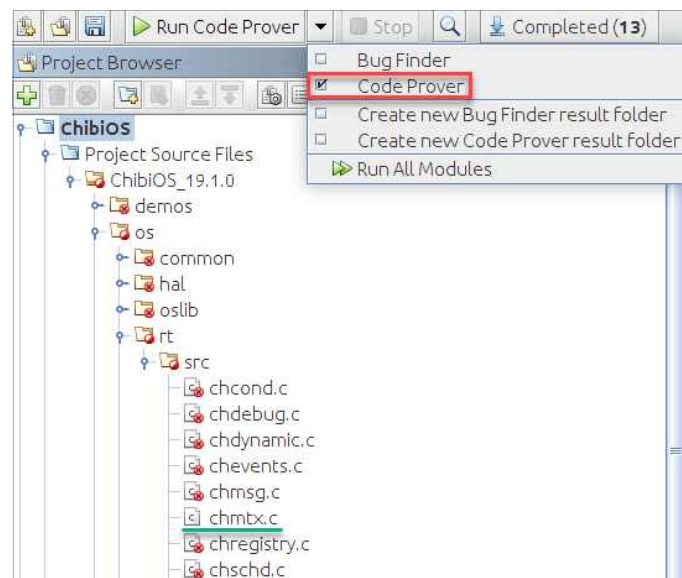


그림 11. 파일 선택 및 Code Prover 실행

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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 체크를 보유하고 있으며, 미탐(False Negatives)이 존재하지 않는다. Bug Finder 는 의미 분석을 포함한 정적 분석을 사용하여 소프트웨어 흐름, 제어 및 동작을 분석하여 결함을 검출 할 수 있다. 결함이 검출되면 알람을 주어 버그를 개발 초기에 파악, 수정할 수 있으나, 오탐이 발생 할 수 있다.

	Bug Finder	Code Prover
Searches for	Software defects	Run-time errors
Check types	More	Less
Analysis depth	Low	High
Exhaustive analysis	No	Yes
Run time scaling	Linear	Exponential
Execution context control	Low	High
Ignorable checks	Yes	No

그림 12. Bug Finder, Code Prover 설명

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

VI. 코드 안전성 분석결과

가. ChibiOS CodeProver 수행결과

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

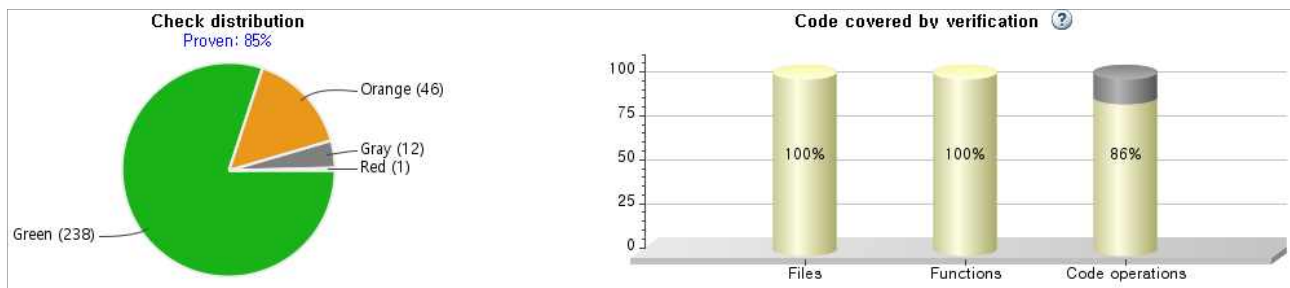


그림 13. ChibiOS chmtx.c 정적분석 결과 개요

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
chsched.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. chsched.h 파일은 Red에서 Illegally dereferenced pointer 1개, Orange에서 Illegally dereferenced pointer 5개, Non-initialized pointer 7개로 총 12개의 약점이 존재한다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

표 2. ChibiOS chsched.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 체크에 의해 ChibiOS chsched.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 체크에 의해 ChibiOS chsched.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.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			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)

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

표 5. ChibiOS chsched.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 chsched.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)

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

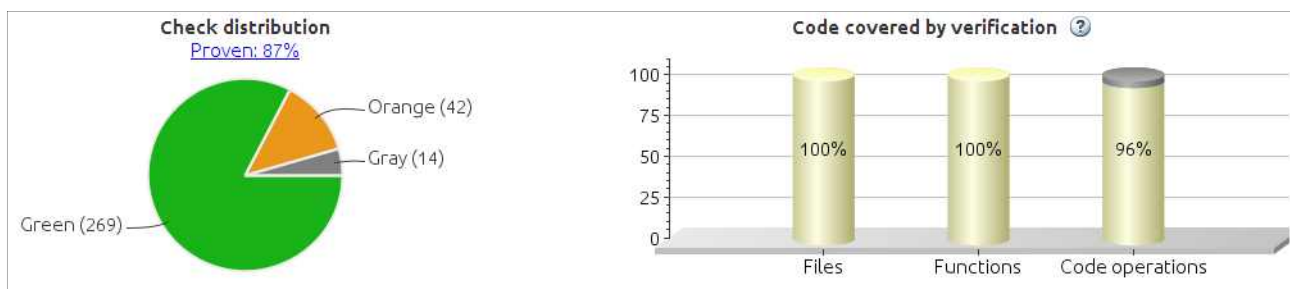


그림 15. CHAOS chmtx.c 정적분석 결과 개요

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	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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

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

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

나. CHAOS version 0.1 CodeProver 수행결과

Polyspace Code Prover를 사용해 CHAOS v0.1의 CHAOS/RT 부분을 정적분석 수행하고 결과를 정리한다.

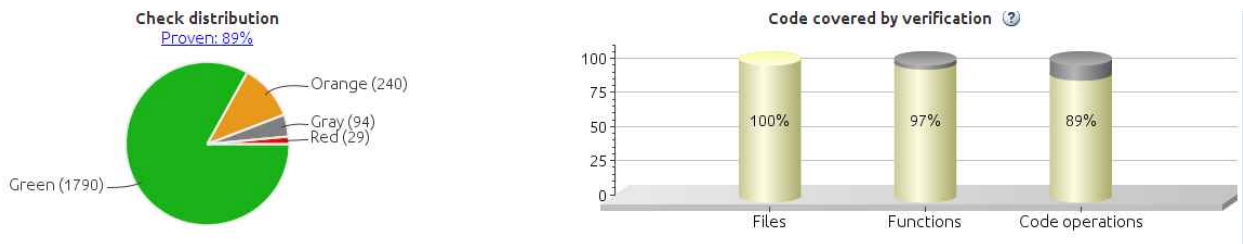


그림 17. CHAOS v0.1 code prover 개요

그림 은 CHAOS/RT 부분의 정적 분석 결과를 나타낸 것이며, Red 29개, Gray 94개, Orange 240개 Green 1790개가 도출되었다.

표 10. CHAOS v0.1 code prover 결과

Color	Check	Count
RED	Return Value Not Initialized	1
	Non Terminating Call	28
GRAY	Unreachable Code	94
ORANGE	Correctness Condition	1
	Illegally Dereferenced Pointer	92
	Invalid Shift Operations	1
	Invalid Use of Standard Library Routine	1
	Non Initialized Local Variable	1
	Non Initialized Pointer	55
	Non Initialized Variable	47
	Overflow	15
	Return Value Not Initialized	27

표10은 CHAOS/RT 정적분석 결과 도출된 약점을 정리한 것 이다. RED에서 Return Value Not Initialized, Non Initialized Pointer 약점이 발견되었고, GRAY에서 Unreachable Code가 발견 되었다. ORANGE에서는 Correctness Condition, Illegally Dereferenced Pointer, Invalid Shift Operations, Invalid Use of Standard Library Routine, Non Initialized Local Variable, Non Initialized Pointer, Non Initialized Variable, Overflow, Return Value Not Initialized의 약점 항목 들이 발견되었고, 총 363개의 약점이 발견 되었다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

표 11. RED 알람

Check	Location	Line	Detail
Return Value Not Initialized	main.c main()	688	Error: function returns a non-initialized value
Non Terminating Call	chthreads.c chThdCreate()	345	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chthreads.c chThdExitS()	571	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsem.c chSemSignal()	303	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chsem.c chSemResetI()	159	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsched.c chSchWakeupS()	1140	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsched.c chSchReadyI()	909	The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.
	chmtx.c chMtxUnlockAllS()	501	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chmsg.c chMsgSend()	96	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chevents.c	309	The called function

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

	chEvtSignal()		chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chdynamic.c chThdCreateFromHeap()	111	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chcond.c chCondBroadcastI()	158	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chcond.c chCondSignal()	95	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chsched.c queue_prio_insert()	740	The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.
	chmsg.h chMsgReleaseS()	117	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.

표 11은 그림 17의 Red 알람의 일부이다. 호출되어 실행되지 않은 함수 약점과 초기화를 하지 않은 반환값이 존재하는 약점이 발생하였다.

표 12. GRAY 알람

Check	Location	Line	Detail
Unreachable code	chdynamic.c chThdCreateFromHeap()	89	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 89 (column 6). Block ends at line 91 (column 2)
	chthreads.h chThdDoDequeueNextI()	445	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 445 (column 11). Block ends

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			at line 445 (column 74)
chcond.c chCondSignal()	91	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 91 (column 11). Block ends at line 91 (column 72)	
chthreads.c chThdWait()	620	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 620 (column 11). Block ends at line 620 (column 72)	
chschd.c storetoKernelArea()	664	The section of code is unreachable or the condition is redundant. If-condition always evaluates to true at line 664 (column 6).	
chmsg.c chMsgRelease()	148	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 148 (column 11). Block ends at line 148 (column 85)	
chevents.c chEvtSignal()	277	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 277 (column 11). Block ends at line 277 (column 72)	
chsem.c chSemObjectInit()	99	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 99 (column 11). Block ends at line 99 (column 93)	
chvt.c chVTDoSetI()	101	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 101 (column 11). Block ends at line 101 (column 135)	
chmtx.c chMtxUnlock()	327	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false	

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			at line 327 (column 11). Block ends at line 327 (column 72)
	chregistry.c chRegNextThread()	170	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 170 (column 13). Block ends at line 170 (column 81)
	chsys.h 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)

표12는 그림17 의 Gray 알람 일부이다. Unreachable code 체크에 의해 CHAOS/RT 도달하지 못하는 코드 약점이 도출 되었다.

표 13. ORANGE 알람

Check	Location	Line	Detail
Correctness Condition	chvt.h chVTDotickl()	389	Warning: function pointer may not point to a valid function Pointer may be null. If pointer is not null, functions that may be called: {_idle_thread, _stub_fun_6, wakeup}. Pointer may point to badly typed function: _port_thread_start. - Error if function _port_thread_start is called: wrong number of arguments (call has 1 argument but function expects 2 arguments).
Illegally Dereferenced Pointer	chsched.c setIDSV_sv()	521	Warning: pointer may be outside its bounds Dereference of parameter 'idsv' (pointer to structure, size: 64 bits): Pointer is not null. Points to 8 bytes at unknown offset in buffer of 8 or 16 bytes, so may be outside bounds. Pointer may point to variable or

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			field of variable: 'ch'. 'idsv', local to function 'enqueue_IDSv'.
Invalid Shift Operations	chevents.c chEvtDispatch()	369	Warning: scalar shift amount may be outside its bounds[0..31] operator << on type unsigned int 32 left: 1 right: [0 .. 32] result: [1 .. 2] or [4 .. 231]
Invalid Use of Standard Library Routine	chregistry.c chRegFindThreadByName()	200	Invalid use of standard library routine Warning: function 'strcmp' is called with possibly invalid argument(s) ? first argument (first string) may not be a valid string Actual value of first argument (first string) (pointer to const int 8, size: 8 bits): pointer may be null. Actual value of second argument (second string) (pointer to const int 8, size: 8 bits): pointer is not null. ? second argument (second string) may not be a valid string Actual value of size of first argument (first string) (int 32): [0 .. 231-1] ✓ value returned fits in range of returned type Actual value of size of second argument (second string) (int 32): [0 .. 67108862 (0x3FFFFFFE)] ✓ value returned fits in range of returned type This check may be an issue related to unbounded input values If appropriate, applying DRS to name (argument number 1 of function chRegFindThreadByName, defined in chregistry.c line 194 and called by

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			main generator), may remove this orange.
Non Initialized Local Variable	chthreads.c enqueue_IDSV()	633	Warning: local variable may be non-initialized Local variable 'idsv': field idsv.id: full-range [0 .. 232-1] field idsv.sv[]: full-range [0 .. 255] Some fields are not used. Check for initialization done only for fields that are used anywhere in code.
Non Initialized Pointer	chsched.c getTlist_next()	196	Warning: pointer may be non-initialized Field of structure (pointer to structure, size: 1088 bits): Pointer may be null. Points to 136 bytes at unknown offset in buffer of unknown size, so may be outside bounds. Pointer may point to dynamically allocated memory. Pointer may point to variable or field of variable: <string_literal>, '_idle_thread'. '_port_thread_start'...
Non Initialized Variable	chsem.c chSemSignalWait()	398	Warning: variable may be non-initialized (type: int 32) Field of structure (int 32): full-range [-231 .. 231-1]
Overflow	chsem.c chSemWaittimeouts()	272	Warning: operation [-] on scalar may overflow (result strictly lower than MIN INT32) This check may be an issue related to unbounded input values If appropriate, applying DRS to sp (argument number 1 of function chSemWaitTimeout, defined in chsem.c line 235 and called by main generator), may remove this orange. operator - on type int 32 left: full-range [-231 .. 231-1]

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			right: 1 result: [-231 .. 2147483646 (0x7FFFFFFE)] (result is truncated)
Return Value Not Initialized	chsched.c firstprio()	178	Warning: function may return a non-initialized value. Returned value of getTQueue_next (pointer to structure, size: 1088 bits): Pointer is not null (but may not be allocated memory). Points to 136 bytes at unknown offset in buffer of unknown size, so may be outside bounds. Pointer may point to dynamically allocated memory. Pointer may point to variable or field of variable: <string_literal>, '_idle_thread'., '_port_thread_start'.

표 13은 그림 17의 ORANGE 알람 일부이다. ORANGE 알람에서는 Correctness Condition, Illegally Dereferenced Pointer, Invalid Shift Operations, Invalid Use of Standard Library Routine, Non Initialized Local Variable, Non Initialized Pointer, Non Initialized Variable, Out of Bounds Array Index, Overflow의 약점이 도출되었다.

※ 자세한 내용은 <부록 1>을 참고

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

다. CHAOS version 0.1 BugFinder 수행결과

Polyspace Bug Finder를 사용해 CHAOS v0.1의 CHAOS/RT 부분을 정적분석 수행하고 결과를 정리한다.

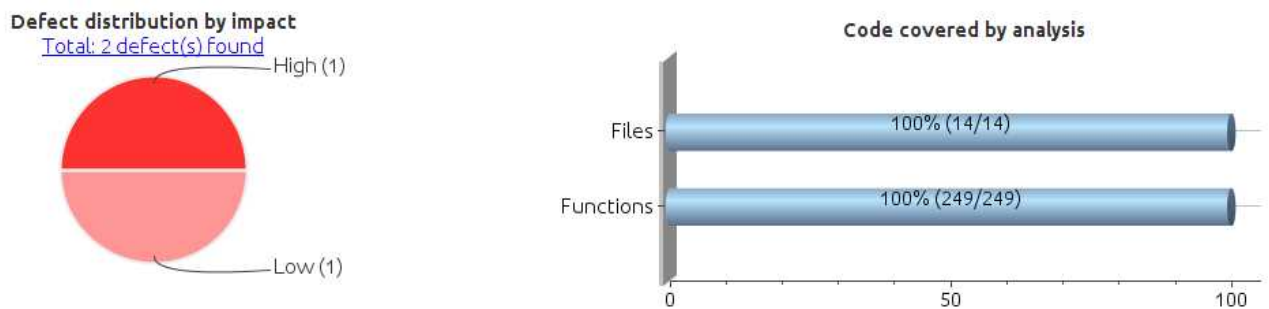


그림 18. CHAOS v0.1 Bug Finder 개요

그림18 은 CHAOS/RT 부분의 Bug Finder 결과를 나타낸 것이며, HIGH 1개, LOW 1개가 도출되었다.

표 14. CHAOS v0.1 Bug Finder 결과

Color	Check	Count
HIGH	Pointer Access Out of Bounds	1
LOW	Missing Return Statement	1

표14 는 CHAOS/RT 파일별 Bug Finder 결과이다. HIGH에서 Pointer Access Out of Bounds 약점이 발견되었고, LOW 에서 Missing Return Statement 약점이 발견 되었고 HIGH에서 1개 LOW에서 1개 가 발견되어 총 2개가 발견되었다.

표 15. HIGH 알람

Check	Location	Line	Detail
Pointer Access Out of Bounds	chschd.c setIDSV_sv()	518	Local pointer 'tp' is read before being initialized.

표15 는 그림18 의 HIGH 알람이다. HIGH 알람에서는 Pointer Access Out of Bounds 의 약점이 도출되었다.

표 16. LOW 알람

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

Check	Location	Line	Detail
Missing Return Statement	chschd.c find_IDSV_in_IDQueue() ue()	704	Missing return statement in non-void function 'find_IDSV_in_IDQueue'.

표16 은 그림18 의 LOW 알람이다. LOW 알람에서는 Missing Return Statement 의 약점이 도출되었다.

※ 자세한 내용은 <부록 2>을 참고

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

라. CHAOS version 0.2 CodeProver 수행결과

Polyspace Code Prover를 사용해 CHAOS v0.2의 CHAOS/RT 부분을 정적분석 수행하고 결과를 정리한다.

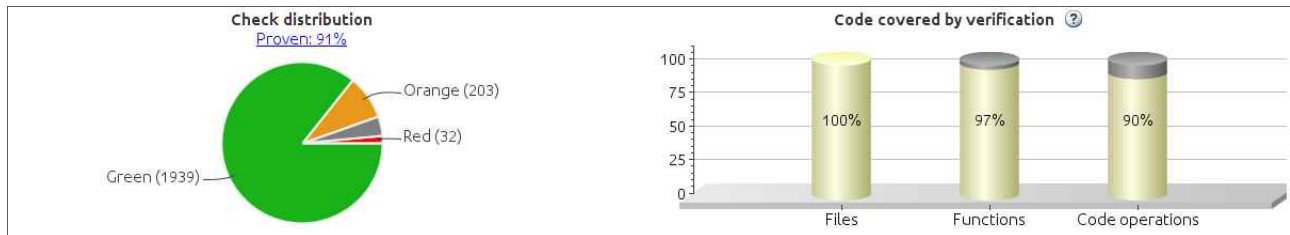


그림 19. CHAOS v0.2 Code Prover 개요

그림 19는 CHAOS/RT 부분의 정적 분석 결과를 나타낸 것이며, Red 32개, Gray 86개, Orange 203개 Green 1939개가 도출되었다.

표 17. CHAOS v0.2 Code Prover 결과

Color	Check	Count
RED	Non Initialized Pointer	2
	Non Terminating Call	30
GRAY	Unreachable Code	86
ORANGE	Correctness Condition	1
	Illegally Dereferenced Pointer	71
	Invalid Shift Operations	1
	Invalid Use of Standard Library Routine	1
	Non Initialized Local Variable	1
	Non Initialized Pointer	55
	Non Initialized Variable	50
	Out of Bounds Array Index	8
	Overflow	15

표17은 CHAOS/RT 파일별 정적분석 결과이다. RED에서 Non Initialized Pointer, Non Terminating Call 약점이 발견되었고, GRAY에서 Unreachable Code가 발견되었다. ORANGE에서는 Correctness Condition, Illegally Dereferenced Pointer, Invalid Shift Operations, Invalid Use of Standard Library Routine, Non Initialized Local Variable, Non Initialized Pointer, Non Initialized Variable, Out of Bounds Array Index, Overflow의 약점 항목들이 발견되었고, 총 321개의 약점이 발견 되었다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

표 18. RED 알람

Check	Location	Line	Detail
Non Initialized Pointer	chthreads.c chThdCreateSuspended()	256	Error: pointer is not initialized
	chthreads.c chThdCreateStatic()	478	Error: pointer is not initialized
Non Terminating Call	chthreads.c chThdStart()	497	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chthreads.c chThdExitS()	637	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsem.c chSemSignal()	303	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chsem.c chSemSignalWait()	391	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsched.c wakeup()	918	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chsched.c chSchReadyI()	772	The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.
	chmtx.c chMtxUnlockS()	468	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chmsg.c chMsgSend()	96	The called function chsched.chSchReadyI (in the

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			current context) either contains an error or does not terminate.
	chevents.c chEvtSignal()	319	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chdynamic.c chThdCreateFromMemoryPool()	177	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chcond.c chCondSignal()	119	The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
	chcond.c chCondSignal()	95	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
	chsched.c queue_prio_insert()	709	The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.
	chmsg.h chMsgReleaseS()	117	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.

표 18은 그림 19의 Red 알람의 일부이다. 호출되어 실행되지 않은 함수 약점과 포인터를 초기화 하지 않은 약점이 발생하였다.

표 19. GRAY 알람

Check	Location	Line	Detail
Unreachable code	chdynamic.c chThdCreateFromMemoryPool()	151	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 151 (column 11). Block ends at line 151 (column 72)
	chthreads.h	377	The section of code is unreachable or

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

	chThdSleeps()		the condition is redundant. If-condition always evaluates to true at line 377 (column 6).
	chcond.c chCondWaitS()	211	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 211 (column 11). Block ends at line 211 (column 72)
	chfia.c fia_generate_sv()	130	The section of code is unreachable or the condition is redundant. If-condition always evaluates to true at line 130 (column 6).
	chthreads.c chThdSuspendT imeoutS()	910	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 910 (column 11). Block ends at line 910 (column 74)
	chsched.c chSchWakeupS()	986	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 986 (column 13). Block ends at line 986 (column 307)
	chmsg.c chMsgRelease()	148	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 148 (column 11). Block ends at line 148 (column 85)
	chevents.c chEvtSignal()	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)
	chsem.c chSemObjectInit ()	99	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 99 (column 11). Block ends at line 99 (column 93)
	chvt.c	101	The section of code is unreachable or

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

	chVTDSetl()		the condition is redundant. If-condition always evaluates to false at line 101 (column 11). Block ends at line 101 (column 135)
	chmtx.c chMtxObjectInit())	105	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 101 (column 11). Block ends at line 101 (column 135)
	chregistry.c chRegNextThread()	170	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 170 (column 13). Block ends at line 170 (column 81)
	chsys.c chSysUnlock()	549	The section of code is unreachable or the condition is redundant. If-condition always evaluates to false at line 549 (column 11). Block ends at line 549 (column 184)

표 19는 그림 19의 Gray 알람 일부이다. Unreachable code चे커에 의해 CHAOS/RT 도달하지 못하는 코드 약점이 도출 되었다.

표 20. ORANGE 알람

Check	Location	Line	Detail
Correctness Condition	chvt.h chVTDSetl()	389	Warning: function pointer may not point to a valid function Pointer may be null. If pointer is not null, functions that may be called: {_idle_thread, _stub_fun_8, wakeup}. Pointer may point to badly typed function: _port_thread_start. - Error if function _port_thread_start is called: wrong number of arguments (call has 1 argument but function expects 2 arguments).
Illegally Dereferenced	chevents.c chEvtSignal()	312	Warning: pointer may be outside its bounds This check may be an issue

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

Pointer			<p>related to unbounded input values If appropriate, applying DRS to tp (argument number 1 of function chEvtSignal, defined in chevents.c line 283 and called by main generator), may remove this orange. Dereference of parameter 'tp' (pointer to structure, size: 1280 bits): Pointer may be null. Points to 160 bytes at unknown offset in buffer of unknown size, so may be outside bounds. Pointer may point to dynamically allocated memory. Pointer may point to variable or field of variable: <string_literal>. '_idle_thread', '_stub_fun_8'...</p>
Invalid Shift Operations	chevents.c chEvtDispatch()	384	<p>Warning: scalar shift amount may be outside its bounds[0..31] operator << on type unsigned int 32 left: 1 right: [0 .. 32] result: [1 .. 2] or [4 .. 231]</p>
Invalid Shift Use of Standard Library Routine	chregistry.c chRegFindThreadByName()	200	<p>Invalid use of standard library routine Warning: function 'strcmp' is called with possibly invalid argument(s) ? first argument (first string) may not be a valid string Actual value of first argument (first string) (pointer to const int 8, size: 8 bits): pointer may be null. Actual value of second argument (second string) (pointer to const int 8, size: 8 bits): pointer is not null. ? second argument (second string) may not be a valid string Actual value of size of first argument (first string) (int 32): [0 .. 231-1]</p>

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

			<p>✓ value returned fits in range of returned type Actual value of size of second argument (second string) (int 32): [0 .. 67108862 (0x3FFFFFFE)] ✓</p> <p>value returned fits in range of returned type This check may be an issue related to unbounded input values If appropriate, applying DRS to name (argument number 1 of function chRegFindThreadByName, defined in chregistry.c line 194 and called by main generator), may remove this orange.</p>
Non Initialized Local Variable	chthreads.c chThdWait()	713	Warning: local variable may be non-initialized (type: int 32) Local variable 'msg' (int 32): full-range [-231 .. 231-1]
Non Initialized Pointer	chsched.c wakeup()	897	<p>Warning: pointer may be non-initialized Field of structure (pointer to pointer, size: 64 bits): Pointer may be null.</p> <p>Points to 8 bytes at unknown offset in buffer of unknown size, so may be outside bounds.</p> <p>Pointer may point to dynamically allocated memory. Pointer may point to variable or field of variable: <string_literal>. '_idle_thread'. '_port_thread_start'...</p>
Non Initialized Variable	chaudit.c auditbox_check_write_authorize()	83	Warning: variable may be non-initialized (type: unsigned int 32) Field of structure (unsigned int 32): full-range [0 .. 232-1]
Out of Bounds Array Index	chfia.c fia_store_idsv_to_kernel_area()	175	Warning: array index may be outside bounds : [0..2047] array size: 2048 array index value: [0 .. 2048]

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

Overflow	chsem.c chSemWaitTimeoutS()	272	Warning: operation [-] on scalar may overflow (result strictly lower than MIN INT32) This check may be an issue related to unbounded input values If appropriate, applying DRS to sp (argument number 1 of function chSemWaitTimeout, defined in chsem.c line 235 and called by main generator), may remove this orange. operator - on type int 32 left: full-range [-231 .. 231-1] right: 1 result: [-231 .. 2147483646 (0x7FFFFFFE)] (result is truncated)
----------	--------------------------------	-----	--

표 20은 그림 19의 ORANGE 알람 일부이다. ORANGE 알람에서는 Correctness Condition, Illegally Dereferenced Pointer, Invalid Shift Operations, Invalid Use of Standard Library Routine, Non Initialized Local Variable, Non Initialized Pointer, Non Initialized Variable, Out of Bounds Array Index, Overflow의 약점이 도출되었다.

※ 자세한 내용은 <부록 3>을 참고

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

마. CHAOS version 0.2 BugFinder 수행결과

Polyspace Bug Finder를 사용해 CHAOS v0.2의 CHAOS/RT 부분을 분석하고 결과를 정리한다.

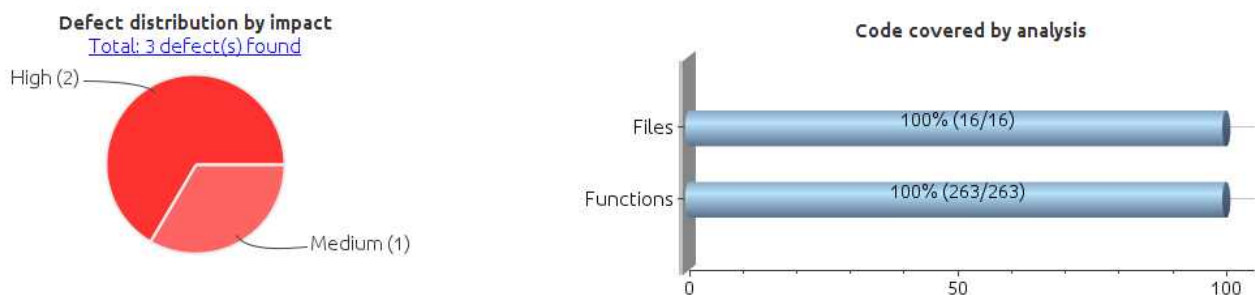


그림 20. CHAOS v0.2 Bug Finder 개요

그림 20은 CHAOS/RT 부분의 Bug Finder 결과를 나타낸 것이며, HIGH 2개, MEDIUM 1개가 도출되었다.

표 21. CHAOS v0.2 Bug Finder 결과

Color	Check	Count
HIGH	Non Initialized Pointer	2
MEDIUM	Incorrect Pointer Scaling	1

표21 은 CHAOS/RT 파일별 Bug Finder 결과이다. HIGH에서 Non Initialized Pointer 약점이 발견되었고, MEDIUM 에서 Incorrect Pointer Scaling 약점이 발견 되었고 HIGH에서 2개 MEDIUM에서 1개 가 발견되어 총 3개가 발견되었다.

표 22. HIGH 알람

Check	Location	Line	Detail
Non Initialized Pointer	chthreads.c chThdCreateSuspended()	256	Local pointer 'tp' is read before being initialized.
	chthreads.c chThdCreateStatic()	478	Local pointer 'tp' is read before being initialized.

표22 는 그림20 의 HIGH 알람이다. HIGH 알람에서는 Non Initialized Pointer 의 약점이 도출

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

되었다.

표 23. MEDIUM 알람

Check	Location	Line	Detail
Incorrect Pointer Scaling	chthreads.c chThdCreateSuspended()	235	Use of 'sizeof' is incorrect because pointer arithmetic with non-char* type is implicitly scaled.

표23 은 그림20 의 MEDIUM 알람이다. MEDIUM 알람에서는 Non Initialized Pointer 의 약점이 도출되었다.

※ 자세한 내용은 <부록 4>을 참고

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

<부록 1> CHAOS version 0.1 CodeProver 결과 분석

분석 결과 아래의 함수들과 관련하여 에러가 발생. 모두 스케줄(chsched.c) 관련 함수

1. thread_t *chSchReadyI(SecretValue kValue, thread_t *tp) - chsched.c 886 Line

2. void setIDSV_sv(IDSV *idsv, SecretValue *sv) - chsched.c 516 Line

It's an individual opinion.

IDQueue_bool_t RegistProcessID(SecretValue kValue)

void setIDSV_sv(IDSV *idsv, SecretValue *sv)

Structure: IDSV

SecreValue sv[SECRET_LEN]

ProcessID 는 하나의 파라미터.

SecreValue 는 두 개의 구조체로 되어있습니다.

3. thread_t *getTQueue_next(threads_queue_t * tq) - chsched.c 206 Line

다른 함수에서 hread_t *getTQueue_next(threads_queue_t * tq)

포인터 함수를 호출하고 반환 인자 값을 받는 과정에 에러가 발생합니다.

4. void chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg) - chsched.c 1113 Line

다른 함수에서 chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg) 함수를 호출하는 과정에 파라미터 값 오류가 다수 발생합니다.

참고

```
#define secret_kValue 0
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

1. chthread.c

분석결과(C)



```

void chThdExitS(SecretValue kValue, msg_t msg)
{
    thread_t *tp = getCurrp();

    /* Storing exit message.*/
    tp->u.exitcode = msg;

    /* Exit handler hook.*/
    CH_CFG_THREAD_EXIT_HOOK(tp);

#ifdef CH_CFG_USE_WAITEXIT == TRUE
    /* Waking up any waiting thread.*/
    while (list_notempty(getThread_waiting(tp))) {
        (void) chSchReadyI(kValue, list_remove(getThread_waiting(tp)));
    }
#endif

#ifdef CH_CFG_USE_REGISTRY == TRUE
    /* Static threads with no references are immediately removed from the
       registry because there is no memory to recover.*/
    #if CH_CFG_USE_DYNAMIC == TRUE
        if ((getThread_refs(tp) == (trefs_t)0) &&
            ((getThread_flags(tp) & CH_FLAG_MODE_MASK0) == CH_FLAG_MODE_STATIC0)) {
            REG_REMOVE(tp);
        }
    #else
        if (getThread_refs(tp) == (trefs_t)0) {
            REG_REMOVE(tp);
        }
    #endif
#endif

    /* Going into final state.*/
    chSchGoSleepS(CH_STATE_FINAL0);

    /* The thread never returns here.*/
    chDbgAssert(false, "zombies apocalypse");
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chthread.c 571 Line

3. 함수이름 : chThdExists()

4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

error or does not terminate.

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ⓘ The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chSysInit'	__polyspace_main.c	main()	471
2	Entering function 'chThdCreate'	chsys.c	chSysInit()	186
3	Entering function 'chSchWakeupS'	chthreads.c	chThdCreate()	345
4	● The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.	chthreads.c	chThdCreate()	345

```

thread_t *chThdCreate(SecretValue kSecret, const thread_descriptor_t *tdp)
{
    //printf("chThdCreate start\n");
    thread_t *tp;
    IDQueue_bool_t registeredIDSV = RegistProcessID(kSecret);
    //printf("chThdCreate - bool: %d\n", get_IDQueue_bool_bool(&registeredIDSV));
    //if (get_IDQueue_bool_bool(&registeredIDSV) == TRUE) {

    #if (CH_CFG_USE_REGISTRY == TRUE) && \
        ((CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE))
        chDbgAssert(chRegFindThreadByWorkingArea(getTD_wbase(tp)) == NULL,
            "working area in use");
    #endif

    #if CH_DBG_FILL_THREADS == TRUE
        _thread_memfill((uint8_t *)getTD_wbase(tp),
            (uint8_t *)getTD_wend(tp),
            CH_DBG_STACK_FILL_VALUE);
    #endif

    chSysLock();
    tp = chThdCreateSuspended(kSecret, tdp);
    chSchWakeupS(kSecret, tp, MSG_OK0);
    chSysUnlock();
    //printf("chThdCreate end\n");
    return tp;
}
//
//return tp;
}

```

- Defect : Non-terminating call
- 위치 : /os/rt/src/chthread.c 345 Line
- 함수이름 : chThdCreate()
- Detail : The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
- Dependent Error :
 - Entering function 'chThdCreate' - chsys.c chSysInit() 186 Line

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

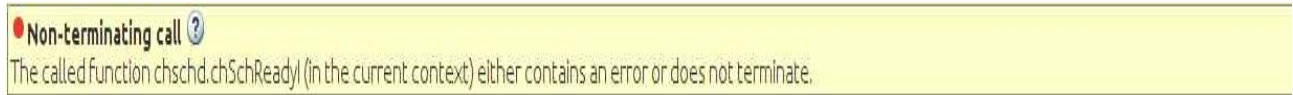
2. Entering function 'chSchWakeupS' - chthread.c chThdCreate() 345 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

2. chsem.c

분석결과(C)



```

msg_t chSemSignalWait(semaphore_t *sps, semaphore_t *spw)
{
    msg_t msg;

    chDbgCheck((sps != NULL) && (spw != NULL));

    chSysLock();
    chDbgAssert(((sps->cnt >= (cnt_t)0) && queue_isempty(&sps->queue)) ||
                ((sps->cnt < (cnt_t)0) && queue_notempty(&sps->queue)),
                "inconsistent semaphore");
    chDbgAssert(((spw->cnt >= (cnt_t)0) && queue_isempty(&spw->queue)) ||
                ((spw->cnt < (cnt_t)0) && queue_notempty(&spw->queue)),
                "inconsistent semaphore");

    if (++sps->cnt <= (cnt_t)0) {
        chSchReadyI(secret_kValue, queue_fifo_remove(&sps->queue))->u.rdymsg = MSG_OK0;
    }
    if (--spw->cnt < (cnt_t)0) {
        thread_t *ctp = getCurtp();
        sem_insert(ctp, &spw->queue);
        ctp->u.wtsemp = spw;
        chSchGoSleep(CH_STATE_WTSEM0);
        msg = ctp->u.rdymsg;
    }
    else {
        chSchRescheduleS0();
        msg = MSG_OK0;
    }
    chSysUnlock();

    return msg;
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chsem.c 391 Line

3. 함수이름 : chSemSignalWait()

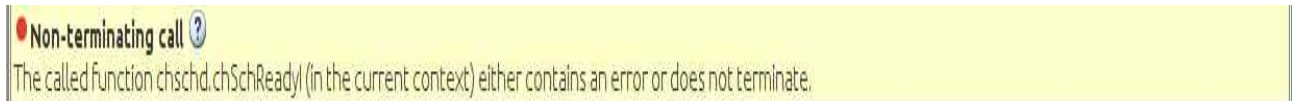
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)



```

void chSemSignal(semaphore_t *sp)
{
    chDbgCheckClass();
    chDbgCheck(sp != NULL);
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

    if (++sp->cnt <= (cnt_t)0) {
        /* Note, it is done this way in order to allow a tail call on
           chSchReadyI().*/
        thread_t *tp = queue_fifo_remove(&sp->queue);
        tp->u.rdymsg = MSG_OK;
        (void) chSchReadyI(secret_kValue, tp);
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chsem.c 332 Line
3. 함수이름 : chSemSignal()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?			
The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
	Event	File	Scope Line
1	Entering function 'chSemSignal'	__polyspace_main.c	main() 437
2	Entering function 'chSchWakeupS'	chsem.c	chSemSignal() 303
3	• The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chsem.c	chSemSignal() 303

```

void chSemSignal(semaphore_t *sp)
{
    chDbgCheck(sp != NULL);

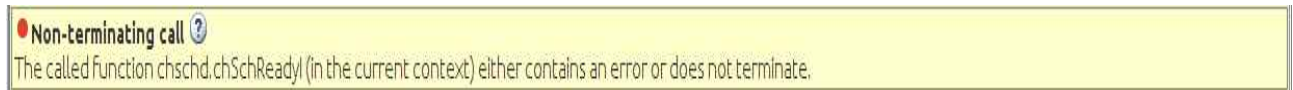
    chSysLock();
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");
    if (++sp->cnt <= (cnt_t)0) {
        chSchWakeupS(secret_kValue, queue_fifo_remove(&sp->queue), MSG_OK0);
    }
    chSysUnlock();
}

```

- Defect : Non-terminating call
- 위치 : /os/rt/src/chsem.c 303 Line
- 함수이름 : chSemSignal()
- Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.
- Dependent Error :
 - Entering function 'chSemSignal' - chsem.c chsemSignal() 303 Line
- Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)



```

void chSemResetI(semaphore_t *sp, cnt_t n)
{
    cnt_t cnt;

    chDbgCheckClass();
    chDbgCheck((sp != NULL) && (n >= (cnt_t)0));
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

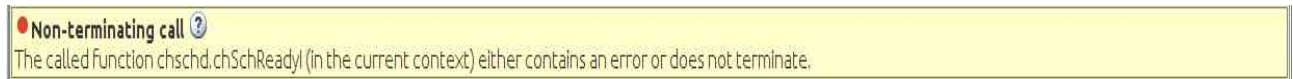
    cnt = sp->cnt;
    sp->cnt = n;
    while (++cnt <= (cnt_t)0) {
        chSchReadyI(secret_kValue, queue_lifo_remove(&sp->queue))->u.rdymsg = MSG_RESET0;
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chsem.c 159 Line
3. 함수이름 : chSemResetI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)



```
void chSemAddCounterI(semaphore_t *sp, cnt_t n)
{
    chDbgCheckClassI();
    chDbgCheck((sp != NULL) && (n > (cnt_t)0));
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

    while (n > (cnt_t)0) {
        if (++sp->cnt <= (cnt_t)0) {
            chSchReadyI(secret_kValue, queue_fifo_remove(&sp->queue))->u.rdymsg = MSG_OK0;
        }
        n--;
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chsem.c 359 Line
3. 함수이름 : chSemAddCounterI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

3. chsched.c

분석결과(C)

Non-terminating call
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
static void wakeup(void *p)
{
    thread_t *tp = (thread_t *)p;

    chSysLockFromISR();
    switch (getThread_state(tp)) {
    case (tstate_t)0:
        /* Handling the special case where the thread has been made ready by
        another thread with higher priority.*/
        chSysUnlockFromISR();
        return;
    case (tstate_t)3:
        *tp->u.wttrp = NULL;
        break;
    #if CH_CFG_USE_SEMAPHORES == TRUE
    case (tstate_t)5:
        chSemFastSignal(tp->u.wtsemp);
    #endif
        /* Falls through.*/
    case (tstate_t)4:
        /* Falls through.*/
    #if (CH_CFG_USE_CONDVARS == TRUE) && (CH_CFG_USE_CONDVARS_TIMEOUT == TRUE)
    case (tstate_t)7:
    #endif
        /* States requiring dequeuing.*/
        (void) queue_dequeue(tp);
        break;
    default:
        /* Any other state, nothing to do.*/
        break;
    }
    setThread_rdymsg(tp, MSG_TIMEOUT);
    //tp->u.rdymsg = MSG_TIMEOUT;
    (void) chSchReadyI(secret_kValue, tp);
    chSysUnlockFromISR();
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chsched.c 1051 Line

3. 함수이름 : wakeup()

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

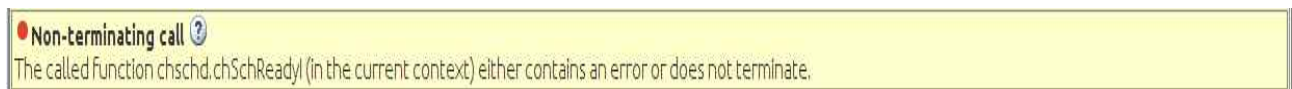
	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)



```

void chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg)
{
    //printf("chSchWakeupS start\n");
    thread_t *otp = getCurp0;
    if (kValue == KernelSecret0) {
        chDbgCheckClassS0;

        chDbgAssert((getTQueue_next(getRlist_queue(getRlist0)) == (thread_t *)getRlist_queue(getRlist0)) &&
            (getThread_prio(getRlist_current(getRlist0)) >= getThread_prio(getTQueue_next(getRlist_queue(getRlist0)))),
            "priority order violation");

        /* Storing the message to be retrieved by the target thread when it will
           restart execution.*/
        //ntp->u.rdymsg = msg;
        setThread_rdymsg(ntp, msg);
        /* If the waken thread has a not-greater priority than the current
           one then it is just inserted in the ready list else it made
           running immediately and the invoking thread goes in the ready
           list instead.*/
        if (getThread_prio(ntp) <= getThread_prio(otp)) {
            (void) chSchReadyI(secret_kValue, ntp);
        }
        else {
            otp = chSchReadyI(secret_kValue, otp);
            /* Handling idle-leave hook.*/
            if (getThread_prio(otp) == IDLEPRIO0) {
                CH_CFG_IDLE_LEAVE_HOOK0;
            }

            /* The extracted thread is marked as current.*/
            setCurp(ntp);
            setThread_state(ntp, CH_STATE_CURRENT0);
            /* Swap operation as tail call.*/
            chSysSwitch(ntp, otp);
        }
    }
    //printf("chSchWakeupS - end\n");
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/src/chsched.c 1132 Line

3. 함수이름 : chSchWakeupS0

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg)
{
    //printf("chSchWakeupS start\n");
    thread_t *otp = getCurtp();
    if (kValue == KernelSecret0) {
        chDbgCheckClassS0;

        chDbgAssert((getTQueue_next(getRlist_queue(getRlist0)) == (thread_t *)getRlist_queue(getRlist0)) ||
            (getThread_prio(getRlist_current(getRlist0)) >= getThread_prio(getTQueue_next(getRlist_queue(getRlist0)))),
            "priority order violation");

        /* Storing the message to be retrieved by the target thread when it will
            restart execution.*/
        //ntp->u.rdymsg = msg;
        setThread_rdymsg(ntp, msg);
        /* If the waken thread has a not-greater priority than the current
            one then it is just inserted in the ready list else it made
            running immediately and the invoking thread goes in the ready
            list instead.*/
        if (getThread_prio(ntp) <= getThread_prio(otp)) {
            (void) chSchReadyI(secret_kValue, ntp);
        }
        else {
            otp = chSchReadyI(secret_kValue, otp);
            /* Handling idle-leave hook.*/
            if (getThread_prio(otp) == IDLEPRIO0) {
                CH_CFG_IDLE_LEAVE_HOOK0;
            }

            /* The extracted thread is marked as current.*/
            setCurtp(ntp);
            setThread_state(ntp, CH_STATE_CURRENT0);
            /* Swap operation as tail call.*/
            chSysSwitch(ntp, otp);
        }
    }
    //printf("chSchWakeupS - end\n");
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/src/chsched.c 1135 Line

3. 함수이름 : chSchWakeupS0

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call			
The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Pointer is outside its bounds	chsched.c	getTQueue_next()
2	The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.	chsched.c	chSchReadyI()

thread_t *chSchReadyI(SecretValue kValue, thread_t *tp)

```
{
    if (kValue == KernelSecret0) {
        //printf("chSchReadyI Start\n");
        thread_t *cp;

        chDbgCheckClassI();
        chDbgCheck(tp != NULL);
        chDbgAssert((getThread_state(tp) != CH_STATE_READY0) &&
                    (getThread_state(tp) != CH_STATE_FINAL0),
                    "invalid state");

        //tp->state = CH_STATE_READY0;
        setThread_state(tp, CH_STATE_READY0);
        //cp = (thread_t *)&ch.rlist.queue;
        cp = (thread_t *)getRlist_queue(getRlist());
        do {
            //cp = cp->queue.next;
            cp = getTQueue_next(getThread_queue(cp));
        } while (cp->prio >= tp->prio);
    } while ((getThread_prio(cp) >= getThread_prio(tp)));
    /* Insertion on prev.*/
    tp->queue.next = cp;
    tp->queue.prev = cp->queue.prev;
    tp->queue.prev->queue.next = tp;
    cp->queue.prev = tp;
    //setTQueue_next(getThread_queue(tp), cp);
    //setTQueue_prev(getThread_queue(tp), getTQueue_prev(getThread_queue(cp)));
    //setTQueue_next(getThread_queue(getTQueue_prev(getThread_queue(tp))), tp);
    //setTQueue_prev(getThread_queue(cp), tp);
    //printf("chSchReadyI end\n");
    return tp;
}
else {
    //printf("chSchReadyI false\n");
    return tp;
}
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsched.c 904 Line
3. 함수이름 : chSchReadyI()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. pointer is outside its bounds - chsched.c getTQueue_next() 207 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ? The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Entering function 'chSchDoReschedule'	__polyspace_main.c	main()
2	Entering function 'chSchReadyAheadl'	chsched.c	chSchDoReschedule()
3	Iterating on loop: loop entered	chsched.c	chSchReadyAheadl()
4	Entering function 'getTQueue_next'	chsched.c	chSchReadyAheadl()
5	● The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.	chsched.c	chSchReadyAheadl()

```
thread_t *chSchReadyAheadl(SecretValue kValue, thread_t *tp)
```

```
{
    if (kValue == KernelSecret()) {
        thread_t *cp;

        chDbgCheckClass();
        chDbgCheck(tp != NULL);
        chDbgAssert((getThread_state(tp) != CH_STATE_READY()) &&
                    (getThread_state(tp) != CH_STATE_FINAL),
                    "invalid state");

        //tp->state = CH_STATE_READY;
        setThread_state(tp, CH_STATE_READY);
        cp = (thread_t *)&ch.rlist.queue;
        do {
            //cp = cp->queue.next;
            cp = getTQueue_next(getThread_queue(cp));
            // while (cp->prio > tp->prio);
        } while ((getThread_prio(cp) > getThread_prio(tp)));
        /* Insertion on prev.*/
        //tp->queue.next = cp;
        //tp->queue.prev = cp->queue.prev;
        //tp->queue.prev->queue.next = tp;
        //cp->queue.prev = tp;
        setTQueue_next(getThread_queue(tp), cp);
        setTQueue_prev(getThread_queue(tp), getTQueue_prev(getThread_queue(cp)));
        setTQueue_next(getThread_queue(getTQueue_prev(getThread_queue(tp))), tp);
        setTQueue_prev(getThread_queue(cp), tp);

        return tp;
    }
    else {
        return tp;
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsched.c 957 Line
3. 함수이름 : chSchReadyAheadl()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.

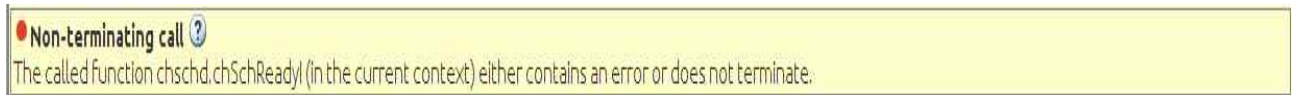
5. Dependent Error :

1. Entering function 'chSchReadyAheadl' - chsched.c chSchDoReschedule() 1306 Line
2. Iterating on loop : loop entered - chsched.c chSchReadyAheadl() 957 Line
3. Entering function 'getTQueue_next - chsched.c chSchReadyAheadl() 957 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)



```

void chSchDoRescheduleBehind(void)
{
    thread_t *otp = getCurrrp0;

    /* Picks the first thread from the ready queue and makes it current.*/
    setCurrrp(queue_fifo_remove(getRlist_queue(getRlist())));
    setThread_state(getCurrrp0, CH_STATE_CURRENT0);

    /* Handling idle-leave hook.*/
    if (getThread_prio(otp) == IDLEPRIO0) {
        CH_CFG_IDLE_LEAVE_HOOK0;
    }

    #if CH_CFG_TIME_QUANTUM > 0
        /* It went behind peers so it gets a new time quantum.*/
        setThread_ticks(otp, (tslices_t)CH_CFG_TIME_QUANTUM);
        //otp->ticks = (tslices_t)CH_CFG_TIME_QUANTUM;
    #endif

    /* Placing in ready list behind peers.*/
    otp = chSchReadyI(secret_kValue, otp);

    /* Swap operation as tail call.*/
    chSysSwitch(getCurrrp0, otp);
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsched.c 1229 Line
3. 함수이름 : chSchDoRescheduleBehind()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating loop ? The loop is infinite or contains a run-time error. Loop fails due to a run-time error (maximum number of iterations: 2).				
	Event	File	Scope	Line
1	Iterating on loop: loop ran 1 time	chschd.c	setIDSV_sv()	517
2	Pointer is outside its bounds	chschd.c	setIDSV_sv()	518
3	● The loop is infinite or contains a run-time error.	chschd.c	setIDSV_sv()	517



```
void setIDSV_sv(IDSV *idsv, SecretValue *sv) {
    for (int i=0;i<SECRET_LEN;i++) {
        idsv->sv[i] = sv[i];
    }
}
```

1. Defect : Non-terminating loop
2. 위치 : os/rt/src/chschd.c 517 Line
3. 함수이름 : setIDSV_sv()
4. Detail : The loop is infinite or contains a run-time error. Loop fails due to a run-time error (maximum number of iterations: 2).
5. Dependent Error :
 1. Iterating on loop : loop ran 1 time - chschd.c setIDSV_sv() 517 Line
 2. Pointer is outside its bounds - chschd.c setIDSV_sv() 518 Line
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. chmtx.c

분석결과(C)

 **Non-terminating call** 
The called function chsched.chSchReady! (in the current context) either contains an error or does not terminate.

```
void chMtxUnlockS(mutex_t *mp)
{
    thread_t *ctp = getCurtp();
    mutex_t *lmp;

    chDbgCheckClassS0;
    chDbgCheck(mp != NULL);

    chDbgAssert(ctp->mtxlist != NULL, "owned mutexes list empty");
    chDbgAssert(ctp->mtxlist->owner == ctp, "ownership failure");
    #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
        chDbgAssert(mp->cnt >= (cnt_t)1, "counter is not positive");

        if (--mp->cnt == (cnt_t)0) {
            #endif

            chDbgAssert(ctp->mtxlist == mp, "not next in list");

            /* Removes the top mutex from the thread's owned mutexes list and marks
             it as not owned. Note, it is assumed to be the same mutex passed as
             parameter of this function.*/
            ctp->mtxlist = mp->next;

            /* If a thread is waiting on the mutex then the fun part begins.*/
            if (chMtxQueueNotEmptyS(mp)) {
                thread_t *tp;

                /* Recalculates the optimal thread priority by scanning the owned
                 mutexes list.*/
                tprio_t newprio = ctp->realprio;
                lmp = ctp->mtxlist;
                while (lmp != NULL) {
                    /* If the highest priority thread waiting in the mutexes list has a
                     greater priority than the current thread base priority then the
                     final priority will have at least that priority.*/
                    if (chMtxQueueNotEmptyS(lmp) &&
                        (lmp->queue.next->prio > newprio)) {
                        newprio = lmp->queue.next->prio;
                    }
                    lmp = lmp->next;
                }

                /* Assigns to the current thread the highest priority among all the
                 waiting threads.*/
                ctp->prio = newprio;

                /* Awakens the highest priority thread waiting for the unlocked mutex and
                 assigns the mutex to it.*/
                #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

    mp->cnt = (cnt_t)1;
#endif
    tp = queue_fifo_remove(&mp->queue);
    mp->owner = tp;
    mp->next = tp->mtxlist;
    tp->mtxlist = mp;
    (void) chSchReadyI(secret_kValue, tp);
}
else {
    mp->owner = NULL;
}
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
}
#endif
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 464 Line
3. 함수이름 : chMtxUnlockSO
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chMtxUnlockAllS(void)
{
    thread_t *ctp = getCurtp();

    while (ctp->mtxlist != NULL) {
        mutex_t *mp = ctp->mtxlist;
        ctp->mtxlist = mp->next;
        if (chMtxQueueNotEmptyS(mp)) {
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                mp->cnt = (cnt_t)1;
            #endif
            thread_t *tp = queue_fifo_remove(&mp->queue);
            mp->owner = tp;
            mp->next = tp->mtxlist;
            tp->mtxlist = mp;

            (void) chSchReadyI(secret_kValue, tp);
        }
        else {
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                mp->cnt = (cnt_t)0;
            #endif
            mp->owner = NULL;
        }
    }
    ctp->prio = ctp->realprio;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 501 Line
3. 함수이름 : chMtxUnlockAllS()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chMtxUnlockAll(void)
{
    thread_t *ctp = getCurtp0;

    chSysLock0;
    if (ctp->mtxlist != NULL) {
        do {
            mutex_t *mp = ctp->mtxlist;
            ctp->mtxlist = mp->next;
            if (chMtxQueueNotEmptyS(mp)) {
                #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                    mp->cnt = (cnt_t)1;
                #endif
                thread_t *tp = queue_fifo_remove(&mp->queue);
                mp->owner = tp;
                mp->next = tp->mtxlist;
                tp->mtxlist = mp;
                (void) chSchReadyI(secret_kValue, tp);
            }
            else {
                #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                    mp->cnt = (cnt_t)0;
                #endif
                mp->owner = NULL;
            }
        } while (ctp->mtxlist != NULL);
        ctp->prio = ctp->realprio;
        chSchRescheduleS0;
    }
    chSysUnlock0;
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 540 Line
3. 함수이름 : chMtxUnlockAll()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call

The called function chsched.chSchReady! (in the current context) either contains an error or does not terminate.

```

void chMtxUnlock(mutex_t *mp)
{
    thread_t *ctp = getCurtp();
    mutex_t *lmp;

    chDbgCheck(mp != NULL);

    chSysLock();

    chDbgAssert(ctp->mtxlist != NULL, "owned mutexes list empty");
    chDbgAssert(ctp->mtxlist->owner == ctp, "ownership failure");
    #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
        chDbgAssert(mp->cnt >= (cnt_t)1, "counter is not positive");
    #endif

    if (--mp->cnt == (cnt_t)0) {
        #endif

        chDbgAssert(ctp->mtxlist == mp, "not next in list");

        /* Removes the top mutex from the thread's owned mutexes list and marks
           it as not owned. Note, it is assumed to be the same mutex passed as
           parameter of this function.*/
        ctp->mtxlist = mp->next;

        /* If a thread is waiting on the mutex then the fun part begins.*/
        if (chMtxQueueNotEmptyS(mp)) {
            thread_t *tp;

            /* Recalculates the optimal thread priority by scanning the owned
               mutexes list.*/
            tprio_t newprio = ctp->realprio;
            lmp = ctp->mtxlist;
            while (lmp != NULL) {
                /* If the highest priority thread waiting in the mutexes list has a
                   greater priority than the current thread base priority then the
                   final priority will have at least that priority.*/
                if (chMtxQueueNotEmptyS(lmp) &&
                    (lmp->queue.next->prio > newprio)) {
                    newprio = lmp->queue.next->prio;
                }
                lmp = lmp->next;
            }

            /* Assigns to the current thread the highest priority among all the
               waiting threads.*/
            ctp->prio = newprio;

            /* Awakens the highest priority thread waiting for the unlocked mutex and
               assigns the mutex to it.*/
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                mp->cnt = (cnt_t)1;
            #endif
        }
    }
}

```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

#endif
    tp = queue_fifo_remove(&mp->queue);
    mp->owner = tp;
    mp->next = tp->mtxlist;
    tp->mtxlist = mp;

    /* Note, not using chSchWakeupS0 because that function expects the
       current thread to have the higher or equal priority than the ones
       in the ready list. This is not necessarily true here because we
       just changed priority.*/
    (void) chSchReadyI(secret_kValue, tp);
    chSchRescheduleS0();
}
else {
    mp->owner = NULL;
}
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
}
#endif

chSysUnlock0;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 383 Line
3. 함수이름 : chMtxUnlock0
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?

The called function chschd.chSchReady! (in the current context) either contains an error or does not terminate.

void chMtxLockS(mutex_t *mp)

```

{
    thread_t *ctp = getCurtp();

    chDbgCheckClassS0;
    chDbgCheck(mp != NULL);

    /* Is the mutex already locked? */
    if (mp->owner != NULL) {
#ifdef CH_CFG_USE_MUTEXES_RECURSIVE == TRUE

        chDbgAssert(mp->cnt >= (cnt_t)1, "counter is not positive");

        /* If the mutex is already owned by this thread, the counter is increased
           and there is no need of more actions.*/
        if (mp->owner == ctp) {
            mp->cnt++;
        }
        else {
#endif
            /* Priority inheritance protocol; explores the thread-mutex dependencies
               boosting the priority of all the affected threads to equal the
               priority of the running thread requesting the mutex.*/
            thread_t *tp = mp->owner;

            /* Does the running thread have higher priority than the mutex
               owning thread? */
            while (tp->prio < ctp->prio) {
                /* Make priority of thread tp match the running thread's priority.*/
                tp->prio = ctp->prio;

                /* The following states need priority queues reordering.*/
                switch (tp->state) {
                    case (tstate_t)6:
                        /* Re-enqueues the mutex owner with its new priority.*/
                        queue_prio_insert(queue_dequeue(tp), &tp->u.wtmtp->queue);
                        tp = tp->u.wtmtp->owner;
                        /*lint -e9042 [16.1] Continues the while.*/
                        continue;
                }
            }
#ifdef CH_CFG_USE_CONDVARS == TRUE
            \
            ((CH_CFG_USE_SEMAPHORES == TRUE) &&
             (CH_CFG_USE_SEMAPHORES_PRIORITY == TRUE)) \
            ((CH_CFG_USE_MESSAGES == TRUE) &&
             (CH_CFG_USE_MESSAGES_PRIORITY == TRUE))
#endif
            \
            #if CH_CFG_USE_CONDVARS == TRUE
                case (tstate_t)7:
            #endif
            \
            #if (CH_CFG_USE_SEMAPHORES == TRUE) &&
                (CH_CFG_USE_SEMAPHORES_PRIORITY == TRUE)
            #endif
                case (tstate_t)5:
            #endif

```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

#if (CH_CFG_USE_MESSAGES == TRUE) && (CH_CFG_USE_MESSAGES_PRIORITY == TRUE)
    case (tstate_t)12:
#endif
    /* Re-enqueues tp with its new priority on the queue.*/
    queue_prio_insert(queue_dequeue(tp), &tp->u.wmtx->queue);
    break;
#endif
    case (tstate_t)0:
#if CH_DBG_ENABLE_ASSERTS == TRUE
    /* Prevents an assertion in chSchReadyI().*/
    tp->state = CH_STATE_CURRENT;
#endif
    /* Re-enqueues tp with its new priority on the ready list.*/
    (void) chSchReadyI(secret_kValue, queue_dequeue(tp));
    break;
default:
    /* Nothing to do for other states.*/
    break;
}
break;
}

/* Sleep on the mutex.*/
queue_prio_insert(ctp, &mp->queue);
ctp->u.wmtx = mp;
chSchGoSleep(CH_STATE_WTMX);

/* It is assumed that the thread performing the unlock operation assigns
the mutex to this thread.*/
chDbgAssert(mp->owner == ctp, "not owner");
chDbgAssert(ctp->mtxlist == mp, "not owned");
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt == (cnt_t)1, "counter is not one");
}
#endif
}
else {
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt == (cnt_t)0, "counter is not zero");

    mp->cnt++;
#endif
    /* It was not owned, inserted in the owned mutexes list.*/
    mp->owner = ctp;
    mp->next = ctp->mtxlist;
    ctp->mtxlist = mp;
}
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 202 Line
3. 함수이름 : chMtxLockS()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				



5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. chmsg.c

분석결과(C)

 **Non-terminating call** 
The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```

msg_t chMsgSend(thread_t *tp, msg_t msg)
{
    thread_t *ctp = getCurtp();

    chDbgCheck(tp != NULL);

    chSysLock();
    ctp->u.sentmsg = msg;
    msg_insert(ctp, &ctp->msgqueue);
    if (ctp->state == CH_STATE_WTMSGO) {
        (void) chSchReadyI(secret_kValue, tp);
    }
    chSchGoSleep(CH_STATE_SNDMSGQO);
    msg = ctp->u.rdymsg;
    chSysUnlock();

    return msg;
}

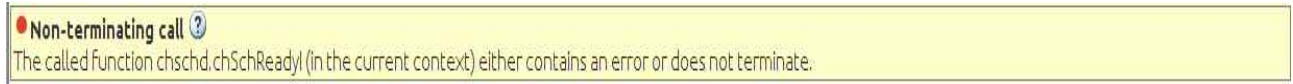
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmsg.c 96 Line
3. 함수이름 : chMsgSend()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

6. chevents.c

분석결과(C)



```

void chEvtSignal(thread_t *tp, eventmask_t events)
{
    chDbgCheckClass();
    chDbgCheck(tp != NULL);

    tp->depending != events;
    /* Test on the AND/OR conditions wait states.*/
    if (((tp->state == CH_STATE_WTOREVT)) &&
        ((tp->depending & tp->u.ewmask) != (eventmask_t0)) &&
        ((tp->state == CH_STATE_WTANDEVTO) &&
        ((tp->depending & tp->u.ewmask) == tp->u.ewmask))) {
        tp->u.rdymsg = MSG_OK0;
        (void) chSchReadyI(secret_kValue, tp);
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chevents.c 309 Line
3. 함수이름 : chEvtSignal()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

7. chdynamic.c

분석결과(C)

Non-terminating call ? The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Entering function 'chThdCreateFromMemoryPool'	__polyspace_main.c	main()
2	Entering function 'chSchWakeupS'	chdynamic.c	chThdCreateFromMem...
3	The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chdynamic.c	chThdCreateFromMem...

```
thread_t *chThdCreateFromMemoryPool(memory_pool_t *mp, const char *name,
tprio_t prio, tfunc_t pf, void *arg)
```

```
{
    thread_t *tp;
    void *wsp;

    chDbgCheck(mp != NULL);

    wsp = chPoolAlloc(mp);
    if (wsp == NULL) {
        return NULL;
    }

    thread_descriptor_t td = {
        name,
        wsp,
        (stkalign_t *)((uint8_t *)wsp + mp->object_size),
        prio,
        pf,
        arg
    };

    #if CH_DBG_FILL_THREADS == TRUE
        _thread_memfill((uint8_t *)wsp,
            (uint8_t *)wsp + mp->object_size,
            CH_DBG_STACK_FILL_VALUE);
    #endif

    chSysLock();
    tp = chThdCreateSuspended(secret_kValue, &td);
    tp->flags = CH_FLAG_MODE_MPOOL0;
    tp->mpool = mp;
    chSchWakeupS(secret_kValue, tp, MSG_OK0);
    chSysUnlock();

    return tp;
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chdynamic.c 175 Line
3. 함수이름 : chThdCreateFromMemoryPool()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chSchWakeupS' - chdynamic.c chThdCreateFromMemoryPool()
175 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chThdCreateFromHeap'	__polyspace_main.c	main()	519
2	Entering function 'chSchWakeupS'	chdynamic.c	chThdCreateFromHeap()	111
3	The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
		chdynamic.c	chThdCreateFromHeap()	111

```

thread_t *chThdCreateFromHeap(memory_heap_t *heapp, size_t size,
                                const char *name, tprio_t prio,
                                tfunc_t pf, void *arg)

```

```

{
    thread_t *tp;
    void *wsp;

    wsp = chHeapAllocAligned(heapp, size, PORT_WORKING_AREA_ALIGN);
    if (wsp == NULL) {
        return NULL;
    }

    thread_descriptor_t td = {
        name,
        wsp,
        (stkalign_t *)((uint8_t *)wsp + size),
        prio,
        pf,
        arg
    };

    #if CH_DBG_FILL_THREADS == TRUE
        _thread_memfill((uint8_t *)wsp,
                        (uint8_t *)wsp + size,
                        CH_DBG_STACK_FILL_VALUE);
    #endif

    chSysLock();
    tp = chThdCreateSuspended(secret_kValue, &td);
    tp->flags = CH_FLAG_MODE_HEAP0;
    chSchWakeupS(secret_kValue, tp, MSG_OK0);
    chSysUnlock();
    return tp;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chdynamic.c 111 Line
3. 함수이름 : chThdCreateFromHeap()
4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.
5. Dependent Error :

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				



1. Entering function 'chSchWakeupS' - chdynamic.c chThdCreateFromeHeap()
111 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

8. chcond.c

분석결과(C)

 **Non-terminating call** 
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chCondSignalI(condition_variable_t *cp)
{
    chDbgCheckClassI();
    chDbgCheck(cp != NULL);

    if (queue_notempty(&cp->queue)) {
        thread_t *tp = queue_fifo_remove(&cp->queue);
        tp->u.rdymsg = MSG_OK0;
        (void) chSchReadyI(secret_kValue, tp);
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chcond.c 119 Line
3. 함수이름 : chCondSignalI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ? The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chCondSignal'	_polyspace_main.c	main()	222
2	Entering function 'chSchWakeupS'	chcond.c	chCondSignal()	95
3	The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.	chcond.c	chCondSignal()	95

```

void chCondSignal(condition_variable_t *cp)
{
    chDbgCheck(cp != NULL);

    chSysLock();
    if (queue_notempty(&cp->queue)) {
        chSchWakeupS(secret_kValue, queue_fifo_remove(&cp->queue), MSG_OK0);
    }
    chSysUnlock();
}

```

- Defect : Non-terminating call
- 위치 : /os/rt/src/chcond.c 95 Line
- 함수이름 : chCondSignal()
- Detail : The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.
- Dependent Error :
 - Entering function 'chSchWakeupS' - chcond.c chCondSignal 95 Line
- Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chCondBroadcastI(condition_variable_t *cp)
{
    chDbgCheckClassI();
    chDbgCheck(cp != NULL);

    /* Empties the condition variable queue and inserts all the threads into the
       ready list in FIFO order. The wakeup message is set to @p MSG_RESET in
       order to make a chCondBroadcast() detectable from a chCondSignal().*/
    while (queue_notempty(&cp->queue)) {
        chSchReadyI(secret_kValue, queue_fifo_remove(&cp->queue))->u.rdymsg = MSG_RESET0;
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chcond.c 158 Line
3. 함수이름 : chCondBroadcastI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

9. chsched.h

분석결과(C)

Non-terminating call ?			
The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Pointer is outside its bounds	chschd.c	getTQueue_next()
2	The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.	chschd.h	queue_prio_insert()

```
static inline void queue_prio_insert(thread_t *tp, threads_queue_t *tqp)
{
    thread_t *cp = (thread_t *)tqp;
    do {
        cp = getTQueue_next(getThread_queue(cp));
    } while ((cp != (thread_t *)tqp) && (getThread_prio(cp) >= getThread_prio(tp)));
    //tp->queue.next = cp;
    //tp->queue.prev = cp->queue.prev;
    //tp->queue.prev->queue.next = tp;
    //cp->queue.prev = tp;
    setTQueue_next(getThread_queue(tp), cp);
    setTQueue_prev(getThread_queue(tp), getTQueue_prev(getThread_queue(cp)));
    setTQueue_next(getThread_queue(getTQueue_prev(getThread_queue(tp))), tp);
    setTQueue_prev(getThread_queue(cp), tp);
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/include/chschd.h 740 Line

3. 함수이름 : queue_prio_insert()

4. Detail : The called function chsched.getTQueue_next (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Pointer is outside its bounds - chschd.c getTQueue_next() 207 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

10. chmsg.h

분석결과(C)

Non-terminating call ? The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chMsgRelease'	__polyspace_main.c	main()	353
2	Entering function 'chMsgReleaseS'	chmsg.c	chMsgRelease()	149
3	Entering function 'chSchWakeupS'	chmsg.h	chMsgReleaseS()	117
4	The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
		chmsg.h	chMsgReleaseS()	117

```
static inline void chMsgReleaseS(thread_t *tp, msg_t msg)
{
    chDbgCheckClassS0;

    chSchWakeupS(secret_kValue, tp, msg);
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/include/chmsg.h 117 Line

3. 함수이름 : chMsgReleaseS()

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chMsgReleaseS' - chmsg.c chMsgReleaseS() 117 Line

2. Entering function 'chSchWakeupS' - chmsg.h chMsgReleaseS() 117 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

<부록 2> CHAOS version 0.1 BugFinder 결과 분석

1. 분석결과(B)

Missing return statement (Impact: Low)

Missing return statement in non-void function 'find_IDSV_in_IDQueue'.

	Event	File	Scope	Line
1	Missing return statement	chschd.c	find_IDSV_in_IDQueue()	704

```

bool find_IDSV_in_IDQueue(SecretValue kValue, ID_t id_in, SecretValue sv_in) {
}

```

- Defect : Missing return statement
 - 위치 : _chschd.c_ 704 Line
 - 함수이름 : find_IDSV_in_IDQueue()
 - Detail : Missing return statement in non-void function 'find_IDSV_in_IDQueue'.
 - Comment
- bool find_IDSV_in_IDQueue (SecretValue kValue, ID_t id_in, SecretValue sv_in) { }
- 반환 자료형 bool에 대한 return 값이 없습니다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

2. 분석결과(B)

Pointer access out of bounds (Impact: High) ? ⓘ Attempt to dereference pointer outside of the pointed object at offset 5.				
	Event	File	Scope	Line
1	Entering function 'RegistProcessID'	chthreads.c	chThdCreate()	327
2	Entering function 'storetoKernelArea'	chschd.c	RegistProcessID()	692
3	Entering function 'enqueue_IDSv'	chschd.c	storetoKernelArea()	662
4	Entering function 'setIDSv_sv'	chschd.c	enqueue_IDSv()	629
5	Iterating on loop: loop ran 1 time	chschd.c	setIDSv_sv()	517
6	Pointer access out of bounds	chschd.c	setIDSv_sv()	518

```

void setIDSv_sv(IDSV *idsv, SecretValue *sv) {
    for (int i=0;i<SECRET_LEN;i++) {
        idsv->sv[i] = sv[i];
    }
}

```

1. Defect : Pointer access out of bounds

2. 위치 : _chschd.c_ 518 Line

3. 함수이름 : setIDSv_sv()

4. Detail : Attempt to dereference pointer outside of the pointed object at offset 5.

5. Comment

IDQueue_bool_t RegistProcessID(SecretValuekValue)

void setIDSv_sv(IDSV *idsv,SecretValue *sv)

Structure: IDSV {

SecreValue sv[SECRET_LEN] }

파라메터의 타입이 맞지 않습니다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

<부록 3> CHAOS version 0.2 CodeProver 결과 분석

분석 결과 아래의 함수들과 관련하여 에러가 발생. 스케줄(chsched.c) 관련 함수 및 쓰레드(chthread.c) 관련 함수

1. thread_t *chSchReadyI(SecretValue kValue, thread_t *tp) - chsched.c 886 Line

다른 함수에서 chSchReadyI 함수를 사용할 때, chSchReadyI 함수 내부에 존재하는 getTQueue_next 함수 처리 과정에서 포인터의 범위를 넘어가는 연산을 수행 할 수 있어 에러가 발생합니다.

2. thread_t *getTQueue_next(threads_queue_t * tq) - chsched.c 206 Line

다른 함수에서 thread_t *getTQueue_next(threads_queue_t * tq)

포인터 함수를 호출하고 반환 인자 값을 받는 과정에 에러가 발생합니다.

3. void chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg) - chsched.c 1113 Line

다른 함수에서 chSchWakeupS(SecretValue kValue, thread_t *ntp, msg_t msg) 함수를 호출하는 과정에 파라미터 값 오류가 다수 발생합니다.

4. thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp) - chthread.c 256 Line

반환하는 thread_t 값이 초기화가 되어있지 않아, 기존에 할당되는 무작위 값이 반환되어 오류를 발생 시킬 수 있습니다.

5. thread_t *chThdCreateStatic(SecretValue_t kValue, SecretValue_t kSecret, void *wsp, size_t size, tprio_t prio, tfunc_t pf, void *arg) - chthread.c 478 Line

반환하는 thread_t 값이 초기화가 되어있지 않아, 기존에 할당되는 무작위 값이 반환되어 오류를 발생 시킬 수 있습니다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

1. chthreads.c

분석결과(C)

● Non-initialized pointer ? Error: pointer is not initialized				
	Event	File	Scope	Line
1	Declaration of variable 'tp'	chthreads.c	chThdCreateSuspendedI()	206
2	Not entering if statement (if-condition False)	chthreads.c	chThdCreateSuspendedI()	212
3	● Error: pointer is not initialized	chthreads.c	chThdCreateSuspendedI()	256

```

thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp)
{
    //printf("chThdCreateSuspendedI start\n");
    thread_t *tp;
    IDSV_bool_t registered_idsv;
    registered_idsv = fia_register_process_id(kSecret);
    auditbox_register_id(registered_idsv.id);
    // //printf("chThdCreateSuspendedI - bool: %d\n", get_IDQueue_bool_bool(&registered_idsv));

    if (registered_idsv.bv) { // TODO: get, set (snd registeredIDSV)
        chDbgCheckClassI();
        chDbgCheck(tdp != NULL);
        // chDbgCheck(MEM_IS_ALIGNED(td_get_wbase(tdp), PORT_WORKING_AREA_ALIGN) &&
        //     MEM_IS_ALIGNED(td_get_wend(tdp), PORT_STACK_ALIGN) &&
        //     (td_get_wend(tdp) > td_get_wbase(tdp)) &&
        //     (((size_t)td_get_wend(tdp) - (size_t)td_get_wbase(tdp)) >= THD_WORKING_AREA_SIZE(0)));
        // chDbgCheck((td_get_prio(tdp) <= HIGHPRIO()) && (td_get_funcp(tdp) != NULL));
        chDbgCheck(MEM_IS_ALIGNED(tdp->wbase, PORT_WORKING_AREA_ALIGN) &&
            MEM_IS_ALIGNED(tdp->wend, PORT_STACK_ALIGN) &&
            (tdp->wend > tdp->wbase) &&
            (((size_t)tdp->wend - (size_t)tdp->wbase) >= THD_WORKING_AREA_SIZE(0)));
        chDbgCheck((tdp->prio <= HIGHPRIO()) && (tdp->funcp != NULL));

        /* The thread structure is laid out in the upper part of the thread
        workspace. The thread position structure is aligned to the required
        stack alignment because it represents the stack top.*/

        // make_thread 부분
        tp = (thread_t *)((uint8_t *)tdp->wend -
            MEM_ALIGN_NEXT(sizeof (thread_t), PORT_STACK_ALIGN));

        // TODO: 자료형 확인
        thread_set_start_addr(tp, tdp->wend - sizeof (thread_t));
        thread_set_end_addr(tp, tdp->wend);

        // set_idsv_thread 부분
        thread_set_id_thd(tp, registered_idsv.id);
        thread_set_pass_thd(tp, registered_idsv.sv);

        #if (CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE)
        /* Stack boundary.*/
        // thread_set_wabase(tp, td_get_wbase(tdp));
        tp->wabase = tdp->wbase;
        #endif
    }
}

```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

/* Setting up the port-dependent part of the working area.*/
// PORT_SETUP_CONTEXT(tp, td_get_wbase(tdp), tp, td_get_funcp(tdp), td_get_arg(tdp));
PORT_SETUP_CONTEXT(tp, tdp->wbase, tp, tdp->funcp, tdp->arg);

/* The driver object is initialized but not started.*/
// return _thread_init(tp, td_get_name(tdp), td_get_prio(tdp));
return _thread_init(tp, tdp->name, tdp->prio);
}
return tp; // null thread (TODO 확인)
}

```

1. Defect : Non-initialized pointer
2. 위치 : /os/rt/src/chthread.c 256 Line
3. 함수이름 : chThdCreateSuspendedI()
4. Detail : Error: pointer is not initialized
5. Dependent Error :
 1. Declaration of variable 'tp' chthreads.c - chThdCreateSuspendedI() 206 Line
 2. Not entering if statement (if-condition false) chthreads.c - chThdCreateSuspendedI() 212 Line
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-initialized pointer ? Error: pointer is not initialized				
	Event	File	Scope	Line
1	Declaration of variable 'tp'	chthreads.c	chThdCreateStatic()	408
2	Not entering if statement (if-condition false)	chthreads.c	chThdCreateStatic()	417
3	● Error: pointer is not initialized	chthreads.c	chThdCreateStatic()	478

```
thread_t *chThdCreateStatic(SecretValue_t kValue, SecretValue_t kSecret,
                           void *wsp, size_t size,
                           tprio_t prio, tfunc_t pf, void *arg)
```

```
{
    thread_t *tp;

    //IDQueue_bool_t registered_idsv = register_process_id(kSecret);
    //if (get_IDQueue_bool_bool(&registered_idsv) == TRUE) {

    IDSV_bool_t registered_idsv;
    registered_idsv = fia_register_process_id(kSecret);
    auditbox_register_id(registered_idsv.id);

    if (registered_idsv.bv) { // TODO: get, set (snd registeredIDSV)

        // chDbgCheck((wsp != NULL) &&
        //             MEM_IS_ALIGNED(wsp, PORT_WORKING_AREA_ALIGN) &&
        //             (size >= THD_WORKING_AREA_SIZE(0)) &&
        //             MEM_IS_ALIGNED(size, PORT_STACK_ALIGN) &&
        //             (prio <= HIGHPRIO()) && (pf != NULL));
        chDbgCheck((wsp != NULL) &&
                  MEM_IS_ALIGNED(wsp, PORT_WORKING_AREA_ALIGN) &&
                  (size >= THD_WORKING_AREA_SIZE(0)) &&
                  MEM_IS_ALIGNED(size, PORT_STACK_ALIGN) &&
                  (prio <= HIGHPRIO) && (pf != NULL));

        #if (CH_CFG_USE_REGISTRY == TRUE) && W
            ((CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE))
            chDbgAssert(chRegFindThreadByWorkingArea(wsp) == NULL,
                      "working area in use");
        #endif

        #if CH_DBG_FILL_THREADS == TRUE
            _thread_memfill((uint8_t *)wsp,
                          (uint8_t *)wsp + size,
                          CH_DBG_STACK_FILL_VALUE);
        #endif

        chSysLock();

        /* The thread structure is laid out in the upper part of the thread
           workspace. The thread position structure is aligned to the required
           stack alignment because it represents the stack top.*/
        // make_thread 부분
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

tp = (thread_t *)((uint8_t *)wsp + size -
MEM_ALIGN_NEXT(sizeof (thread_t), PORT_STACK_ALIGN));
thread_set_start_addr(tp, wsp + size - sizeof (thread_t)); // TODO: 자료형 확인
thread_set_end_addr(tp, wsp + size); // TODO: 자료형 확인

// set_idsv_thread 부분
thread_set_id_thd(tp, registered_idsv.id);
thread_set_pass_thd(tp, registered_idsv.sv);

#if (CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE)
/* Stack boundary.*/
// thread_set_wabase(tp, (stkalign_t *)wsp);
tp->wabase = (stkalign_t *)wsp;
#endif

/* Setting up the port-dependent part of the working area.*/
PORT_SETUP_CONTEXT(tp, wsp, tp, pf, arg);

tp = _thread_init(tp, "noname", prio);

/* Starting the thread immediately.*/
chSchWakeupS(kValue, tp, MSG_OK());
chSysUnlock();

// CHAOS audit
// auditbox_try_update_buffer("chThdCreateStatic", true);
auditbox_try_update_buffer(__func__, true); // TODO __func__가 잘 동작하는지 확인

return tp;
}
return tp; // TODO: null thread?
}

```

1. Defect : Non-initialized pointer
2. 위치 : /os/rt/src/chthread.c 478 Line
3. 함수이름 : chThdCreateStatic()
4. Detail : Error: pointer is not initialized
5. Dependent Error :
 1. Declaration of variable 'tp' chthreads.c - chThdCreateStatic() 408 Line
 2. Not entering if statement (if-condition false) chthreads.c - chThdCreateStatic() 417 Line
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ? The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
	Event	File	Scope Line
1	Entering function 'chThdStart'	__polyspace_main.c	main() 706
2	Entering function 'chSchWakeupS'	chthreads.c	chThdStart() 497
3	● The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chthreads.c	chThdStart() 497

```

thread_t *chThdStart(SecretValue_t kValue, thread_t *tp)
{
    chSysLock();
    if (kValue == get_kernel_secret()) {
        if (fia_check_id_sv(kValue, thread_get_id_thd(tp), thread_get_pass_thd(tp))) {
            // chDbgAssert(thread_get_state(tp) == CH_STATE_WTSTART(), "wrong state");
            chDbgAssert(tp->state == CH_STATE_WTSTART(), "wrong state");
            chSchWakeupS(kValue, tp, MSG_OK());
        }
    }

    chSysUnlock();
    auditbox_try_update_buffer(__func__, true);
    return tp;
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chthreads.c 497 Line

3. 함수이름 : chThdStart()

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chThdStart' __polyspace_main.c - main() 706 Line

2. Entering function 'chSchWakeupS' chthreads.c chThdStart() - 497 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

 **Non-terminating call** 
The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chThdExitS(SecretValue_t kValue, msg_t msg)
{
    thread_t *tp = getCurrrp();

    auditbox_remove_id(tp->id_thd); // TODO: check 종료되는 시점이 여기가 마지막인지

    /* Storing exit message.*/
    tp->u.exitcode = msg;

    /* Exit handler hook.*/
    CH_CFG_THREAD_EXIT_HOOK(tp);

    #if CH_CFG_USE_WAITEXIT == TRUE
        /* Waking up any waiting thread.*/
        while (!list_notempty(thread_get_waiting_ptr(tp))) {
            (void) chSchReadyI(kValue, list_remove(thread_get_waiting_ptr(tp)));
        }
    #endif

    #if CH_CFG_USE_REGISTRY == TRUE
        /* Static threads with no references are immediately removed from the
        registry because there is no memory to recover.*/
    #if CH_CFG_USE_DYNAMIC == TRUE
        if ((thread_get_refs(tp) == (trefs_t)0) &&
            ((thread_get_flags(tp) & CH_FLAG_MODE_MASK()) == CH_FLAG_MODE_STATIC())) {
            REG_REMOVE(tp);
        }
    #else
        if (thread_get_refs(tp) == (trefs_t)0) {
            REG_REMOVE(tp);
        }
    #endif
    #endif

    /* Going into final state.*/
    chSchGoSleepS(CH_STATE_FINAL());

    /* The thread never returns here.*/
    chDbgAssert(false, "zombies apocalypse");

    auditbox_try_update_buffer(__func__, true);
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chthreads.c 637 Line
3. 함수이름 : chThdExitS()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ?

The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
thread_t *chThdCreateI(SecretValue_t kValue, const thread_descriptor_t *tdp)
{
    return chSchReadyI(kValue, chThdCreateSuspendedI(kValue, tdp));
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chthreads.c 334 Line
3. 함수이름 : chThdCreateI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ?			
The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Entering function 'chSysinit'	__polyspace_main.c	main()
2	Entering function 'chThdCreate'	chsys.c	chSysinit()
3	Entering function 'chSchWakeupS'	chthreads.c	chThdCreate()
4	● The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chthreads.c	chThdCreate()

thread_t *chThdCreate(SecretValue_t kSecret, const thread_descriptor_t *tdp)

```
{
    //printf("chThdCreate start\n");

    thread_t *tp;

    #if (CH_CFG_USE_REGISTRY == TRUE) &&                                W
        ((CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE))

        // chDbgAssert(chRegFindThreadByWorkingArea(td_get_wbase(tdp)) == NULL,

        //          "working area in use");

        chDbgAssert(chRegFindThreadByWorkingArea(tdp->wbase) == NULL,

        "working area in use");

    #endif

    #if CH_DBG_FILL_THREADS == TRUE

        // _thread_memfill((uint8_t *)td_get_wbase(tdp),

        //          (uint8_t *)td_get_wend(tdp),

        //          CH_DBG_STACK_FILL_VALUE);

        _thread_memfill((uint8_t *)tdp->wbase,

        (uint8_t *)tdp->wend,

        CH_DBG_STACK_FILL_VALUE);

    #endif

    chSysLock();

    tp = chThdCreateSuspendedI(kSecret, tdp);

    chSchWakeupS(kSecret, tp, MSG_OK());

    chSysUnlock();
}
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

auditbox_try_update_buffer(__func__, true); // CHAOS audit

return tp;

//}

//return tp;

}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chthreads.c 376 Line
3. 함수이름 : chThdCreate()
4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.
5. Dependent Error :
 1. Entering function 'chSysInit' __polyspace_main.c - main() 510 Line
 2. Entering function 'chThdCreate' chsys.c - chSysInit() 189 Line
 3. Entering function 'chSchWakeupS' chthreads.c - chThdCreate() 376 Line
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

2. chsem.c

분석결과(C)

 **Non-terminating call** 
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

msg_t chSemSignalWait(semaphore_t *sps, semaphore_t *spw)
{
    msg_t msg;

    chDbgCheck((sps != NULL) && (spw != NULL));

    chSysLock();
    chDbgAssert(((sps->cnt >= (cnt_t)0) && queue_isempty(&sps->queue)) ||
                ((sps->cnt < (cnt_t)0) && queue_notempty(&sps->queue)),
                "inconsistent semaphore");
    chDbgAssert(((spw->cnt >= (cnt_t)0) && queue_isempty(&spw->queue)) ||
                ((spw->cnt < (cnt_t)0) && queue_notempty(&spw->queue)),
                "inconsistent semaphore");
    if (++sps->cnt <= (cnt_t)0) {
        chSchReadyI(0, queue_fifo_remove(&sps->queue))->u.rdymsg = MSG_OK();
    }
    if (--spw->cnt < (cnt_t)0) {
        thread_t *ctp = getCurtp();
        sem_insert(ctp, &spw->queue);
        ctp->u.wtsemp = spw;
        chSchGoSleepS(CH_STATE_WTSEM());
        msg = ctp->u.rdymsg;
    }
    else {
        chSchRescheduleS();
        msg = MSG_OK();
    }
    chSysUnlock();

    return msg;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chsem.c 391 Line
3. 함수이름 : chSemSignalWait()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

 **Non-terminating call** 
The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chSemSignal(semaphore_t *sp)
{
    chDbgCheckClassI();
    chDbgCheck(sp != NULL);
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

    if (++sp->cnt <= (cnt_t)0) {
        /* Note, it is done this way in order to allow a tail call on
           chSchReadyI().*/
        thread_t *tp = queue_fifo_remove(&sp->queue);
        tp->u.rdymsg = MSG_OK();
        (void) chSchReadyI(0, tp);
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chsem.c 332 Line
3. 함수이름 : chSemSignal()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ? The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering Function 'chSemSignal'	__polyspace_main.c	main()	469
2	Entering Function 'chSchWakeupS'	chsem.c	chSemSignal()	303
3	• The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chsem.c	chSemSignal()	303

```

void chSemSignal(semaphore_t *sp)
{
    chDbgCheck(sp != NULL);

    chSysLock();
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");
    if (++sp->cnt <= (cnt_t)0) {
        chSchWakeupS(0, queue_fifo_remove(&sp->queue), MSG_OK());
    }
    chSysUnlock();
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/src/chsem.c 303 Line

3. 함수이름 : chSemSignal()

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chSemSignal' __polyspace_main.c - main() 469 Line

2. Entering function 'chSchWakeupS' - chsem.c chSemSignal() 303 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

 **Non-terminating call** 
The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chSemResetI(semaphore_t *sp, cnt_t n)
{
    cnt_t cnt;

    chDbgCheckClassI();
    chDbgCheck((sp != NULL) && (n >= (cnt_t)0));
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

    cnt = sp->cnt;
    sp->cnt = n;
    while (++cnt <= (cnt_t)0) {
        chSchReadyI(0, queue_lifo_remove(&sp->queue))->u.rdymsg = MSG_RESET();
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsem.c 159 Line
3. 함수이름 : chSemResetI()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chSemResetI(semaphore_t *sp, cnt_t n)
{
    cnt_t cnt;

    chDbgCheckClassI();
    chDbgCheck((sp != NULL) && (n >= (cnt_t)0));
    chDbgAssert(((sp->cnt >= (cnt_t)0) && queue_isempty(&sp->queue)) ||
                ((sp->cnt < (cnt_t)0) && queue_notempty(&sp->queue)),
                "inconsistent semaphore");

    cnt = sp->cnt;
    sp->cnt = n;
    while (++cnt <= (cnt_t)0) {
        chSchReadyI(0, queue_lifo_remove(&sp->queue))->u.rdymsg = MSG_RESET();
    }
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsem.c 359 Line
3. 함수이름 : chSemResetI()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

3. chsched.c

분석결과(C)

 **Non-terminating call** 
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
static void wakeup(void *p)
{
    thread_t *tp = (thread_t *)p;

    chSysLockFromISR();
    switch (thread_get_state(tp)) {
    case (tstate_t)0:
        /* Handling the special case where the thread has been made ready by
           another thread with higher priority.*/
        chSysUnlockFromISR();
        return;
    case (tstate_t)3:
        *tp->u.wttrp = NULL;
        break;
    #if CH_CFG_USE_SEMAPHORES == TRUE
    case (tstate_t)5:
        chSemFastSignal(tp->u.wtsemp);
    #endif
        /* Falls through.*/
    case (tstate_t)4:
        /* Falls through.*/
    #if (CH_CFG_USE_CONDVARS == TRUE) && (CH_CFG_USE_CONDVARS_TIMEOUT == TRUE)
    case (tstate_t)7:
    #endif
        /* States requiring dequeuing.*/
        (void) queue_dequeue(tp);
        break;
    default:
        /* Any other state, nothing to do.*/
        break;
    }
    thread_set_rdymsg(tp, MSG_TIMEOUT());
    //tp->u.rdymsg = MSG_TIMEOUT();
    (void) chSchReadyI(0, tp);
    chSysUnlockFromISR();
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/src/chsched.c 918 Line

3. 함수이름 : wakeup()

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)


Non-terminating call ?

The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chSchWakeupS(SecretValue_t kValue, thread_t *ntp, msg_t msg)
{
    //printf("chSchWakeupS startWn");
    thread_t *otp = getCurp();
    if (kValue == get_kernel_secret()) {
        chDbgCheckClassS();

        chDbgAssert(((tqueue_get_next(rlist_get_queue_ptr(ch_get_rlist_ptr())) == (thread_t *)rlist_get_queue_ptr(ch_get_rlist_ptr())) ||
            (thread_get_prio(rlist_get_current(ch_get_rlist_ptr())) >= thread_get_prio(tqueue_get_next(rlist_get_queue_ptr(ch_get_rlist_ptr()))),
            "priority order violation");

        /* Storing the message to be retrieved by the target thread when it will
        restart execution.*/
        //ntp->u.rdymsg = msg;
        thread_set_rdymsg(ntp, msg);
        /* If the waken thread has a not-greater priority than the current
        one then it is just inserted in the ready list else it made
        running immediately and the invoking thread goes in the ready
        list instead.*/
        if (thread_get_prio(ntp) <= thread_get_prio(otp)) {
            (void) chSchReadyI(0, ntp); //error
        }
        else {
            otp = chSchReadyI(0, otp);
            /* Handling idle-leave hook.*/
            if (thread_get_prio(otp) == IDLEPRIO()) {
                CH_CFG_IDLE_LEAVE_HOOK();
            }

            /* The extracted thread is marked as current.*/
            setCurp(ntp);
            thread_set_state(ntp, CH_STATE_CURRENT());
            /* Swap operation as tail call.*/
            chSysSwitch(ntp, otp);
        }
    }
    //printf("chSchWakeupS - endWn");
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/src/chsched.c 999 Line
3. 함수이름 : chSchWakeupS()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)


Non-terminating call

The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chSchWakeupS(SecretValue_t kValue, thread_t *ntp, msg_t msg)
{
    //printf("chSchWakeupS start\n");
    thread_t *otp = getCurp();
    if (kValue == get_kernel_secret()) {
        chDbgCheckClassS();

        chDbgAssert(((tqueue_get_next(rlist_get_queue_ptr(ch_get_rlist_ptr())) == (thread_t *)rlist_get_queue_ptr(ch_get_rlist_ptr())) ||
            (thread_get_prio(rlist_get_current(ch_get_rlist_ptr())) >= thread_get_prio(tqueue_get_next(rlist_get_queue_ptr(ch_get_rlist_ptr()))),
            "priority order violation");

        /* Storing the message to be retrieved by the target thread when it will
        restart execution.*/
        //ntp->u.rdymsg = msg;
        thread_set_rdymsg(ntp, msg);
        /* If the waken thread has a not-greater priority than the current
        one then it is just inserted in the ready list else it made
        running immediately and the invoking thread goes in the ready
        list instead.*/
        if (thread_get_prio(ntp) <= thread_get_prio(otp)) {
            (void) chSchReadyI(0, ntp);
        }
        else {
            otp = chSchReadyI(0, otp);
            /* Handling idle-leave hook.*/
            if (thread_get_prio(otp) == IDLEPRIO()) {
                CH_CFG_IDLE_LEAVE_HOOK();
            }

            /* The extracted thread is marked as current.*/
            setCurp(ntp);
            thread_set_state(ntp, CH_STATE_CURRENT());
            /* Swap operation as tail call.*/
            chSysSwitch(ntp, otp);
        }
    }
    //printf("chSchWakeupS - end\n");
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/src/chsched.c 1002 Line

3. 함수이름 : chSchWakeupS()

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call The called function chschd.tqueue_get_next (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Pointer is outside its bounds	chschd.c	tqueue_get_next()	207
2	The called function chschd.tqueue_get_next (in the current context) either contains an error or does not terminate.	chschd.c	chSchReadyI()	772

```

thread_t *chSchReadyI(SecretValue_t kValue, thread_t *tp)
{
    if (kValue == get_kernel_secret()) {
        //printf("chSchReadyI Start\n");
        thread_t *cp;

        chDbgCheckClassI();
        chDbgCheck(tp != NULL);
        chDbgAssert((thread_get_state(tp) != CH_STATE_READY()) &&
            (thread_get_state(tp) != CH_STATE_FINAL()),
            "invalid state");

        //tp->state = CH_STATE_READY();
        thread_set_state(tp, CH_STATE_READY());
        //cp = (thread_t *)&ch.rlist.queue;
        cp = (thread_t *)rlist_get_queue_ptr(ch_get_rlist_ptr());
        do {
            //cp = cp->queue.next;
            cp = tqueue_get_next(thread_get_queue_ptr(cp));
            //} while (cp->prio >= tp->prio);
        } while ((thread_get_prio(cp) >= thread_get_prio(tp)));
        /* Insertion on prev.*/

        // tp->queue.next = cp;
        // tp->queue.prev = cp->queue.prev;
        // tp->queue.prev->queue.next = tp;
        // cp->queue.prev = tp;
        tqueue_set_next(thread_get_queue_ptr(tp), cp);
        tqueue_set_prev(thread_get_queue_ptr(tp), tqueue_get_prev(thread_get_queue_ptr(cp)));
        tqueue_set_next(thread_get_queue_ptr(tqueue_get_prev(thread_get_queue_ptr(tp))), tp);
        tqueue_set_prev(thread_get_queue_ptr(cp), tp);
        //printf("chSchReadyI end\n");
        return tp;
    }
    else {
        //printf("chSchReadyI false\n");
        return tp;
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chschd.c 772 Line
3. 함수이름 : chSchReadyI()
4. Detail : The called function chschd.tqueue_get_next (in the current context) either contains

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

an error or does not terminate.

5. Dependent Error :

1. Pointer is outside its bounds chsched.c - tqueue_get_next() 207 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ? The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Entering Function 'chSchDoReschedule'	__polyspace_main.c	main()
2	Entering Function 'chSchReadyAheadl'	chsched.c	chSchDoReschedule()
3	Iterating on loop; loop entered	chsched.c	chSchReadyAheadl()
4	Entering Function 'tqueue_get_next'	chsched.c	chSchReadyAheadl()
5	● The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.	chsched.c	chSchReadyAheadl()

thread_t *chSchReadyAheadl(SecretValue_t kValue, thread_t *tp)

```
{
    if (kValue == get_kernel_secret()) {
        thread_t *cp;

        chDbgCheckClass();
        chDbgCheck(tp != NULL);
        chDbgAssert((thread_get_state(tp) != CH_STATE_READY) &&
                    (thread_get_state(tp) != CH_STATE_FINAL),
                    "invalid state");

        //tp->state = CH_STATE_READY;
        thread_set_state(tp, CH_STATE_READY);
        cp = (thread_t *)&ch.rlist.queue;
        do {
            //cp = cp->queue.next;
            cp = tqueue_get_next(thread_get_queue_ptr(cp));
            // while (cp->prio > tp->prio);
        } while ((thread_get_prio(cp) > thread_get_prio(tp)));
        /* Insertion on prev.*/
        //tp->queue.next = cp;
        //tp->queue.prev = cp->queue.prev;
        //tp->queue.prev->queue.next = tp;
        //cp->queue.prev = tp;
        tqueue_set_next(thread_get_queue_ptr(tp), cp);
        tqueue_set_prev(thread_get_queue_ptr(tp), tqueue_get_prev(thread_get_queue_ptr(cp)));
        tqueue_set_next(thread_get_queue_ptr(tqueue_get_prev(thread_get_queue_ptr(tp))), tp);
        tqueue_set_prev(thread_get_queue_ptr(cp), tp);

        return tp;
    }
    else {
        return tp;
    }
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chsched.c 826 Line

3. 함수이름 : chSchReadyAheadl()

4. Detail : The called function chsched.tqueue_get_next (in the current context) either contains

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

an error or does not terminate.

5. Dependent Error :

1. Entering function 'chSchDoReschedule' __polyspace_main.c - main() 439 Line
2. Entering function 'chSchReadyAheadI' chsched.c - chSchDoReschedule() 1175 Line
3. Iterating on loop: loop entered chsched.c - chSchReadyAheadI() 824 Line
4. Entering function 'tqueue_get_next' chsched.c - chSchReadyAheadI() 826 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

 **Non-terminating call ?**
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

void chSchDoRescheduleBehind(void)

```
{
    thread_t *otp = getCurrp();

    /* Picks the first thread from the ready queue and makes it current.*/
    setCurrp(queue_fifo_remove(rlist_get_queue_ptr(ch_get_rlist_ptr())));
    thread_set_state(getCurrp(), CH_STATE_CURRENT());

    /* Handling idle-leave hook.*/
    if (thread_get_prio(otp) == IDLEPRIO()) {
        CH_CFG_IDLE_LEAVE_HOOK();
    }

    #if CH_CFG_TIME_QUANTUM > 0
    /* It went behind peers so it gets a new time quantum.*/
    thread_set_ticks(otp, (tslices_t)CH_CFG_TIME_QUANTUM);
    //otp->ticks = (tslices_t)CH_CFG_TIME_QUANTUM;
    #endif

    /* Placing in ready list behind peers.*/
    otp = chSchReadyI(0, otp); //error

    /* Swap operation as tail call.*/
    chSysSwitch(getCurrp(), otp);
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chsched.c 1098 Line

3. 함수이름 : chSchDoRescheduleBehind()

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

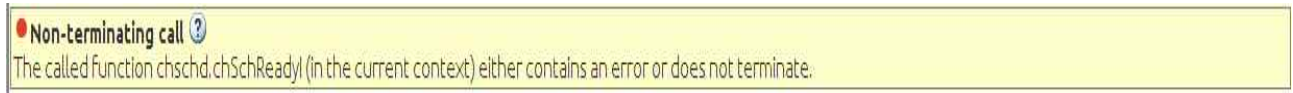
5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. chmtx.c

분석결과(C)



void chMtxUnlockS(mutex_t *mp)

```
{
    thread_t *ctp = getCurp();
    mutex_t *lmp;

    chDbgCheckClassS();
    chDbgCheck(mp != NULL);

    chDbgAssert(ctp->mtxlist != NULL, "owned mutexes list empty");
    chDbgAssert(ctp->mtxlist->owner == ctp, "ownership failure");
    #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt >= (cnt_t)1, "counter is not positive");

    if (--mp->cnt == (cnt_t)0) {
    #endif

        chDbgAssert(ctp->mtxlist == mp, "not next in list");

        /* Removes the top mutex from the thread's owned mutexes list and marks
         it as not owned. Note, it is assumed to be the same mutex passed as
         parameter of this function.*/
        ctp->mtxlist = mp->next;

        /* If a thread is waiting on the mutex then the fun part begins.*/
        if (chMtxQueueNotEmptyS(mp)) {
            thread_t *tp;

            /* Recalculates the optimal thread priority by scanning the owned
             mutexes list.*/
            tprio_t newprio = ctp->realprio;
            lmp = ctp->mtxlist;
            while (lmp != NULL) {
                /* If the highest priority thread waiting in the mutexes list has a
                 greater priority than the current thread base priority then the
                 final priority will have at least that priority.*/
                if (chMtxQueueNotEmptyS(lmp) &&
                    (lmp->queue.next->prio > newprio)) {
                    newprio = lmp->queue.next->prio;
                }
                lmp = lmp->next;
            }

            /* Assigns to the current thread the highest priority among all the
             waiting threads.*/
            ctp->prio = newprio;

            /* Awakens the highest priority thread waiting for the unlocked mutex and
             assigns the mutex to it.*/
        }
    }
}
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```


#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    mp->cnt = (cnt_t)1;
#endif
    tp = queue_fifo_remove(&mp->queue);
    mp->owner = tp;
    mp->next = tp->mtxlist;
    tp->mtxlist = mp;
    (void) chSchReadyI(0, tp);
}
else {
    mp->owner = NULL;
}
#endif CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
}
#endif
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 468 Line
3. 함수이름 : chMtxUnlockS0
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

 **Non-terminating call** ?
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chMtxUnlockAllS(void)
{
    thread_t *ctp = getCurrp0;

    while (ctp->mtxlist != NULL) {
        mutex_t *mp = ctp->mtxlist;
        ctp->mtxlist = mp->next;
        if (chMtxQueueNotEmptyS(mp)) {
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                mp->cnt = (cnt_t)1;
            #endif
            thread_t *tp = queue_fifo_remove(&mp->queue);
            mp->owner = tp;
            mp->next = tp->mtxlist;
            tp->mtxlist = mp;

            (void) chSchReadyI(0, tp);
        }
        else {
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                mp->cnt = (cnt_t)0;
            #endif
            mp->owner = NULL;
        }
    }
    ctp->prio = ctp->realprio;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 505 Line
3. 함수이름 : chMtxUnlockAllS()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```
void chMtxUnlockAll(void)
{
    thread_t *ctp = getCurp();

    chSysLock();
    if (ctp->mtxlist != NULL) {
        do {
            mutex_t *mp = ctp->mtxlist;
            ctp->mtxlist = mp->next;
            if (chMtxQueueNotEmptyS(mp)) {
                #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
                    mp->cnt = (cnt_t)1;
                #endif
                thread_t *tp = queue_fifo_remove(&mp->queue);
                mp->owner = tp;
                mp->next = tp->mtxlist;
                tp->mtxlist = mp;
                (void) chSchReadyI(0, tp); //error
            }
        } while (ctp->mtxlist != NULL);
        ctp->prio = ctp->realprio;
        chSchRescheduleS();
    }
    chSysUnlock();
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 544 Line
3. 함수이름 : chMtxUnlockAll()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

Non-terminating call ?

The called function chSchd.chSchReady! (in the current context) either contains an error or does not terminate.

void chMtxUnlock(mutex_t *mp)

```
{
    thread_t *ctp = getCurp();
    mutex_t *lmp;

    chDbgCheck(mp != NULL);

    chSysLock();

    chDbgAssert(ctp->mtxlist != NULL, "owned mutexes list empty");
    chDbgAssert(ctp->mtxlist->owner == ctp, "ownership failure");
    #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt >= (cnt_t)1, "counter is not positive");

    if (--mp->cnt == (cnt_t)0) {
    #endif

        chDbgAssert(ctp->mtxlist == mp, "not next in list");

        /* Removes the top mutex from the thread's owned mutexes list and marks
         it as not owned. Note, it is assumed to be the same mutex passed as
         parameter of this function.*/
        ctp->mtxlist = mp->next;

        /* If a thread is waiting on the mutex then the fun part begins.*/
        if (chMtxQueueNotEmptyS(mp)) {
            thread_t *tp;

            /* Recalculates the optimal thread priority by scanning the owned
             mutexes list.*/
            tprio_t newprio = ctp->realprio;
            lmp = ctp->mtxlist;
            while (lmp != NULL) {
                /* If the highest priority thread waiting in the mutexes list has a
                 greater priority than the current thread base priority then the
                 final priority will have at least that priority.*/
                if (chMtxQueueNotEmptyS(lmp) &&
                    (lmp->queue.next->prio > newprio)) {
                    newprio = lmp->queue.next->prio;
                }
                lmp = lmp->next;
            }

            /* Assigns to the current thread the highest priority among all the
             waiting threads.*/
            ctp->prio = newprio;

            /* Awakens the highest priority thread waiting for the unlocked mutex and
             assigns the mutex to it.*/
            #if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
            mp->cnt = (cnt_t)1;
            #endif
        }
    }
}
```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

#endif

tp = queue_fifo_remove(&mp->queue);
mp->owner = tp;
mp->next = tp->mtxlist;
tp->mtxlist = mp;

/* Note, not using chSchWakeupS() because that function expects the
current thread to have the higher or equal priority than the ones
in the ready list. This is not necessarily true here because we
just changed priority.*/
(void) chSchReadyI(0, tp);
chSchRescheduleS();
}
else {
mp->owner = NULL;
}
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
}
#endif

chSysUnlock();

auditbox_try_update_buffer(__func__, true); // CHAOS audit
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 385 Line
3. 함수이름 : chMtxUnlock()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

#endif
#if (CH_CFG_USE_MESSAGES == TRUE) && (CH_CFG_USE_MESSAGES_PRIORITY == TRUE)
    case (tstate_t)12:
#endif
    /* Re-enqueues tp with its new priority on the queue.*/
    queue_prio_insert(queue_dequeue(tp), &tp->u.wtmtp->queue);
    break;
#endif
    case (tstate_t)0:
#if CH_DBG_ENABLE_ASSERTS == TRUE
    /* Prevents an assertion in chSchReadyI().*/
    tp->state = CH_STATE_CURRENT;
#endif
    /* Re-enqueues tp with its new priority on the ready list.*/
    (void) chSchReadyI(0, queue_dequeue(tp));
    break;
default:
    /* Nothing to do for other states.*/
    break;
}
break;
}

/* Sleep on the mutex.*/
queue_prio_insert(ctp, &mp->queue);
ctp->u.wtmtp = mp;
chSchGoSleep(CH_STATE_WTMTPX);

/* It is assumed that the thread performing the unlock operation assigns
the mutex to this thread.*/
chDbgAssert(mp->owner == ctp, "not owner");
chDbgAssert(ctp->mtxlist == mp, "not owned");
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt == (cnt_t)1, "counter is not one");
}
#endif
}
else {
#if CH_CFG_USE_MUTEXES_RECURSIVE == TRUE
    chDbgAssert(mp->cnt == (cnt_t)0, "counter is not zero");

    mp->cnt++;
#endif
    /* It was not owned, inserted in the owned mutexes list.*/
    mp->owner = ctp;
    mp->next = ctp->mtxlist;
    ctp->mtxlist = mp;
}
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmtx.c 204 Line
3. 함수이름 : chMtxLockS()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

5. Dependent Error : NONE

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. chmsg.c

분석결과(C)

Non-terminating call ?
The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```
msg_t chMsgSend(thread_t *tp, msg_t msg)
{
    thread_t *ctp = getCurp();

    chDbgCheck(tp != NULL);

    chSysLock();
    ctp->u.sentmsg = msg;
    msg_insert(ctp, &tp->msgqueue);
    if (tp->state == CH_STATE_WTMMSG()) {
        (void) chSchReadyI(0, tp); //error
    }
    chSchGoSleepS(CH_STATE_SNDMSGQ());
    msg = ctp->u.rdymsg;
    chSysUnlock();

    return msg;
}
```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chmsg.c 96 Line
3. 함수이름 : chMsgSend()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

6. chevents.c

분석결과(C)

Non-terminating call ?
The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.

```

msg_t chMsgSend(thread_t *tp, msg_t msg)
{
    thread_t *ctp = getCurp();

    chDbgCheck(tp != NULL);

    chSysLock();
    ctp->u.sentmsg = msg;
    msg_insert(ctp, &tp->msgqueue);
    if (tp->state == CH_STATE_WTMMSG()) {
        (void) chSchReadyI(0, tp); //error
    }
    chSchGoSleepS(CH_STATE_SNDMSGQ());
    msg = ctp->u.rdymsg;
    chSysUnlock();

    return msg;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chevents.c 319 Line
3. 함수이름 : chMsgSend()
4. Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

7. chdynamic.c

분석결과(C)

Non-terminating call				
The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chThdCreateFromMemoryPool'	__polyspace_main.c	main()	585
2	Entering function 'chSchWakeupS'	chdynamic.c	chThdCreateFromMemoryPool()	177
3	The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
		chdynamic.c	chThdCreateFromMemoryPool()	177

```

thread_t *chThdCreateFromMemoryPool(SecretValue_t kValue, memory_pool_t *mp
                                     const char *name, tprio_t prio, tfunc_t pf, void *arg)
{
    thread_t *tp;
    void *wsp;

    chDbgCheck(mp != NULL);

    wsp = chPoolAlloc(mp);
    if (wsp == NULL) {
        return NULL;
    }

    thread_descriptor_t td = {
        name,
        wsp,
        (stkaligned_t *)((uint8_t *)wsp + mp->object_size),
        prio,
        pf,
        arg
    };

    #if CH_DBG_FILL_THREADS == TRUE
        _thread_memfill((uint8_t *)wsp,
                        (uint8_t *)wsp + mp->object_size,
                        CH_DBG_STACK_FILL_VALUE);
    #endif

    chSysLock();
    tp = chThdCreateSuspended(kValue, &td);
    tp->flags = CH_FLAG_MODE_MPOOL();
    tp->mpool = mp;
    chSchWakeupS(kValue, tp, MSG_OK());
    chSysUnlock();

    auditbox_try_update_buffer(__func__, true); // CHAOS audit

    return tp;
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chdynamic.c 177 Line
3. 함수이름 : chThdCreateFromMemoryPool()

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chThdCreateFromMemoryPool' __polyspace_main.c - main() 585 Line

2. Entering function 'chSchWakeupS' chdynamic.c - chThdCreateFromMemoryPool() 177 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ? The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chThdCreateFromHeap'	__polyspace_main.c	main()	569
2	Entering function 'chSchWakeupS'	chdynamic.c	chThdCreateFromHeap()	111
3	● The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chdynamic.c	chThdCreateFromHeap()	111

```

thread_t *chThdCreateFromHeap(SecretValue_t kValue, memory_heap_t *heapp, size_t size,
                               const char *name, tprio_t prio,
                               tfunc_t pf, void *arg)

```

```

{
    thread_t *tp;
    void *wsp;

    wsp = chHeapAllocAligned(heapp, size, PORT_WORKING_AREA_ALIGN);
    if (wsp == NULL) {
        return NULL;
    }

```

```

    thread_descriptor_t td = {
        name,
        wsp,
        (stkalign_t *)((uint8_t *)wsp + size),
        prio,
        pf,
        arg
    };

```

```

#if CH_DBG_FILL_THREADS == TRUE
    _thread_memfill((uint8_t *)wsp,
                    (uint8_t *)wsp + size,
                    CH_DBG_STACK_FILL_VALUE);
#endif

```

```

chSysLock();
tp = chThdCreateSuspended(kValue, &td);
tp->flags = CH_FLAG_MODE_HEAP();
chSchWakeupS(kValue, tp, MSG_OK());
chSysUnlock();

```

```

auditbox_try_update_buffer(__func__, true); // CHAOS audit
return tp;
}

```

1. Defect : Non-terminating call

2. 위치 : /os/rt/src/chdynamic.c 111 Line

3. 함수이름 : chThdCreateFromHeap()

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

5. Dependent Error :

1. Entering function 'chThdCreateFromHeap' __polyspace_main.c - main() 569 Line
2. Entering function 'chSchWakeupS' chdynamic.c - chThdCreateFromHeap() 111 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

8. chcond.c

분석결과(C)


Non-terminating call ?

The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chCondSignal(condition_variable_t *cp)
{
    chDbgCheckClassI();
    chDbgCheck(cp != NULL);

    if (queue_notempty(&cp->queue)) {
        thread_t *tp = queue_fifo_remove(&cp->queue);
        tp->u.rdymsg = MSG_OK();
        (void) chSchReadyI(0, tp);
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chcond.c 119 Line
3. 함수이름 : chCondSignal()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

● Non-terminating call ⓘ The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.				
	Event	File	Scope	Line
1	Entering function 'chCondSignal'	__polyspace_main.c	main()	254
2	Entering function 'chSchWakeupS'	chcond.c	chCondSignal()	95
3	● The called function chsched.chSchWakeupS (in the current context) either contains an error or does not terminate.	chcond.c	chCondSignal()	95

```
void chCondSignal(condition_variable_t *cp)
{
    chDbgCheck(cp != NULL);

    chSysLock();
    if (queue_notempty(&cp->queue)) {
        chSchWakeupS(0, queue_fifo_remove(&cp->queue), MSG_OK());
    }
    chSysUnlock();
}
```

- Defect : Non-terminating call
- 위치 : /os/rt/src/chcond.c 95 Line
- 함수이름 : chCondSignal()
- Detail : The called function chsched.chSchReadyI (in the current context) either contains an error or does not terminate.
- Dependent Error :
 - Entering function 'chCondSignal' __polyspace_main.c - main() 254 Line
 - Entering function 'chSchWakeupS' chcond.c - chCondSignal() 95 Line
- Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)


Non-terminating call


The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.

```

void chCondBroadcastI(condition_variable_t *cp)
{
    chDbgCheckClassI();
    chDbgCheck(cp != NULL);

    /* Empties the condition variable queue and inserts all the threads into the
       ready list in FIFO order. The wakeup message is set to @p MSG_RESET in
       order to make a chCondBroadcast() detectable from a chCondSignal().*/
    while (queue_notempty(&cp->queue)) {
        chSchReadyI(0, queue_fifo_remove(&cp->queue))->u.rdymsg = MSG_RESET(); //error
    }
}

```

1. Defect : Non-terminating call
2. 위치 : /os/rt/src/chcond.c 158 Line
3. 함수이름 : chCondBroadcastI()
4. Detail : The called function chschd.chSchReadyI (in the current context) either contains an error or does not terminate.
5. Dependent Error : NONE
6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

9. chsched.h

분석결과(C)

Non-terminating call ? The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Pointer is outside its bounds	chschd.c	tqueue_get_next() 207
2	The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.	chschd.h	queue_prio_insert() 709

```
static inline void queue_prio_insert(thread_t *tp, threads_queue_t *tqp)
{
    thread_t *cp = (thread_t *)tqp;
    do {
        cp = tqueue_get_next(thread_get_queue_ptr(cp));
    } while ((cp != (thread_t *)tqp) && (thread_get_prio(cp) >= thread_get_prio(tp)));
    //tp->queue.next = cp;
    //tp->queue.prev = cp->queue.prev;
    //tp->queue.prev->queue.next = tp;
    //cp->queue.prev = tp;
    tqueue_set_next(thread_get_queue_ptr(tp), cp);
    tqueue_set_prev(thread_get_queue_ptr(tp), tqueue_get_prev(thread_get_queue_ptr(cp)));
    tqueue_set_next(thread_get_queue_ptr(tqueue_get_prev(thread_get_queue_ptr(tp))), tp);
    tqueue_set_prev(thread_get_queue_ptr(cp), tp);
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/include/chschd.h 709 Line

3. 함수이름 : queue_prio_insert()

4. Detail : The called function chsched.tqueue_get_next (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Pointer is outside its bounds chschd.c - tqueue_get_next() 207 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

10. chmsg.h

분석결과(C)

● Non-terminating call ⓘ The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.			
	Event	File	Scope
1	Entering function 'chMsgRelease'	__polyspace_main.c	main()
2	Entering function 'chMsgReleaseS'	chmsg.c	chMsgRelease()
3	Entering function 'chSchWakeupS'	chmsg.h	chMsgReleaseS()
4	● The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.	chmsg.h	chMsgReleaseS()

```
static inline void chMsgReleaseS(thread_t *tp, msg_t msg)
{
    chDbgCheckClassS0;

    chSchWakeupS(secret_kValue, tp, msg);
}
```

1. Defect : Non-terminating call

2. 위치 : /os/rt/include/chmsg.h 117 Line

3. 함수이름 : chMsgReleaseS()

4. Detail : The called function chschd.chSchWakeupS (in the current context) either contains an error or does not terminate.

5. Dependent Error :

1. Entering function 'chMsgRelease' __polyspace_main.c - main() 385 Line

2. Entering function 'chMsgReleaseS' chmsg.c - chMsgRelease() 149 Line

3. Entering function 'chSchWakeupS' chmsg.h - chMsgReleaseS() 117 Line

6. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보 고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

<부록 4> CHAOS version 0.2 BugFinder 결과 분석

에러가 탐지된 함수는 아래와 같다.

1. `thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp)`

반환하는 `thread_t` 값이 초기화가 되어있지 않아, 기존에 할당되는 무작위 값이 반환되어 오류를 발생 시킬 수 있다.

2. `thread_t *chThdCreateStatic(SecretValue_t kValue, SecretValue_t kSecret, void *wsp, size_t size, tprio_t prio, tfunc_t pf, void *arg)`

반환하는 `thread_t` 값이 초기화가 되어있지 않아, 기존에 할당되는 무작위 값이 반환되어 오류를 발생 시킬 수 있다.

3. `thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp)`

포인터 산술이 암시적으로 크기 조정되기 때문에 'sizeof' 함수를 사용을 해서는 안된다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

1. chthreads.c

분석결과(C)

<div> <div> <div></div> <div>Non-initialized pointer</div> <div>?</div> </div> <div>Error: pointer is not initialized</div> </div>				
	Event	File	Scope	Line
1	Declaration of variable 'tp'	chthreads.c	chThdCreateSuspendedI()	206
2	Not entering if statement (if-condition false)	chthreads.c	chThdCreateSuspendedI()	212
3	● Error: pointer is not initialized	chthreads.c	chThdCreateSuspendedI()	256

```

thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp)
{
    //printf("chThdCreateSuspendedI start\n");
    thread_t *tp;
    IDSV_bool_t registered_idsv;
    registered_idsv = fia_register_process_id(kSecret);
    auditbox_register_id(registered_idsv.id);
    // //printf("chThdCreateSuspendedI - bool: %d\n", get_IDQueue_bool_bool(&registered_idsv));

    if (registered_idsv.bv) { // TODO: get, set (snd registeredIDSV)
        chDbgCheckClass();
        chDbgCheck(tdp != NULL);
        // chDbgCheck(MEM_IS_ALIGNED(td_get_wbase(tdp), PORT_WORKING_AREA_ALIGN) &&
        //     MEM_IS_ALIGNED(td_get_wend(tdp), PORT_STACK_ALIGN) &&
        //     (td_get_wend(tdp) > td_get_wbase(tdp)) &&
        //     (((size_t)td_get_wend(tdp) - (size_t)td_get_wbase(tdp)) >= THD_WORKING_AREA_SIZE(0)));
        // chDbgCheck((td_get_prio(tdp) <= HIGHPRIO()) && (td_get_funcp(tdp) != NULL));
        chDbgCheck(MEM_IS_ALIGNED(tdp->wbase, PORT_WORKING_AREA_ALIGN) &&
            MEM_IS_ALIGNED(tdp->wend, PORT_STACK_ALIGN) &&
            (tdp->wend > tdp->wbase) &&
            (((size_t)tdp->wend - (size_t)tdp->wbase) >= THD_WORKING_AREA_SIZE(0)));
        chDbgCheck((tdp->prio <= HIGHPRIO()) && (tdp->funcp != NULL));

        /* The thread structure is laid out in the upper part of the thread
        workspace. The thread position structure is aligned to the required
        stack alignment because it represents the stack top.*/

        // make_thread 부분
        tp = (thread_t *)(((uint8_t *)tdp->wend -
            MEM_ALIGN_NEXT(sizeof (thread_t), PORT_STACK_ALIGN));

        // TODO: 자료형 확인
        thread_set_start_addr(tp, tdp->wend - sizeof (thread_t));
        thread_set_end_addr(tp, tdp->wend);

        // set_idsv_thread 부분
        thread_set_id_thd(tp, registered_idsv.id);
        thread_set_pass_thd(tp, registered_idsv.sv);

        #if (CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE)
        /* Stack boundary.*/
        // thread_set_wbase(tp, td_get_wbase(tdp));
        tp->wbase = tdp->wbase;
        #endif
    }
}

```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

/* Setting up the port-dependent part of the working area.*/
// PORT_SETUP_CONTEXT(tp, td_get_wbase(tdp), tp, td_get_funcp(tdp), td_get_arg(tdp));
PORT_SETUP_CONTEXT(tp, tdp->wbase, tp, tdp->funcp, tdp->arg);

/* The driver object is initialized but not started.*/
// return _thread_init(tp, td_get_name(tdp), td_get_prio(tdp));
return _thread_init(tp, tdp->name, tdp->prio);
}
return tp; // null thread (TODO 확인)
}

```

1. Defect : Non-initialized pointer
2. Impact : HIGH
3. 위치 : /os/rt/src/chthread.c 256 Line
4. 함수이름 : chThdCreateSuspendedI()
5. Detail : Local pointer 'tp' is read before being initialized.
6. Dependent Error :
 1. Declaration of variable 'tp' chthreads.c - chThdCreateSuspendedI() 206 Line
 2. Not entering if statement (if-condition false) chthreads.c - chThdCreateSuspendedI() 212 Line
7. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

thread_set_start_addr(tp, wsp + size - sizeof (thread_t)); // TODO: 자료형 확인
thread_set_end_addr(tp, wsp + size); // TODO: 자료형 확인

// se_idsv_thread 부분
thread_set_id_thd(tp, registered_idsv.id);
thread_set_pass_thd(tp, registered_idsv.sv);

#if (CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE)
/* Stack boundary.*/
// thread_set_wabase(tp, (stkalgn_t *)wsp);
tp->wabase = (stkalgn_t *)wsp;
#endif

/* Setting up the port-dependent part of the working area.*/
PORT_SETUP_CONTEXT(tp, wsp, tp, pf, arg);

tp = _thread_init(tp, "noname", prio);

/* Starting the thread immediately.*/
chSchWakeupS(kValue, tp, MSG_OK());
chSysUnlock();

// CHAOS audit
// auditbox_try_update_buffer("chThdCreateStatic", true);
auditbox_try_update_buffer(__func__, true); // TODO __func__가 잘 동작하는지 확인

return tp;
}
return tp; // TODO: null thread?
}

```

1. Defect : Non-initialized pointer
2. Impact : HIGH
3. 위치 : /os/rt/src/chthread.c 478 Line
4. 함수이름 : chThdCreateStatic()
5. Detail : Local pointer 'tp' is read before being initialized.
6. Dependent Error :
 1. Declaration of variable 'tp' chthreads.c - chThdCreateStatic() 408 Line
 2. Not entering if statement (if-condition false) chthreads.c - chThdCreateStatic() 417 Line
7. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

분석결과(C)

<p>Incorrect pointer scaling (Impact: Medium) ? ⓘ</p> <p>Use of 'sizeof' is incorrect because pointer arithmetic with non-char* type is implicitly scaled.</p>				
	Event	File	Scope	Line
1	Entering function 'chThdCreateSuspendedI'	chthreads.c	chThdCreate()	375
2	Incorrect pointer scaling	chthreads.c	chThdCreateSuspendedI()	235

```

thread_t *chThdCreateSuspendedI(SecretValue_t kSecret, const thread_descriptor_t *tdp)
{
    //printf("chThdCreateSuspendedI start\n");
    thread_t *tp;
    IDSV_bool_t registered_idsv;
    registered_idsv = fia_register_process_id(kSecret);
    auditbox_register_id(registered_idsv.id);
    // //printf("chThdCreateSuspendedI - bool: %d\n", get_IDQueue_bool_bool(&registered_idsv));

    if (registered_idsv.bv) { // TODO: get, set (snd registeredIDSV)
        chDbgCheckClassI();
        chDbgCheck(tdp != NULL);
        // chDbgCheck(MEM_IS_ALIGNED(td_get_wbase(tdp), PORT_WORKING_AREA_ALIGN) &&
        //     MEM_IS_ALIGNED(td_get_wend(tdp), PORT_STACK_ALIGN) &&
        //     (td_get_wend(tdp) > td_get_wbase(tdp)) &&
        //     (((size_t)td_get_wend(tdp) - (size_t)td_get_wbase(tdp)) >= THD_WORKING_AREA_SIZE(0)));
        // chDbgCheck((td_get_prio(tdp) <= HIGHPRIO()) && (td_get_funcp(tdp) != NULL));
        chDbgCheck(MEM_IS_ALIGNED(tdp->wbase, PORT_WORKING_AREA_ALIGN) &&
            MEM_IS_ALIGNED(tdp->wend, PORT_STACK_ALIGN) &&
            (tdp->wend > tdp->wbase) &&
            (((size_t)tdp->wend - (size_t)tdp->wbase) >= THD_WORKING_AREA_SIZE(0)));
        chDbgCheck((tdp->prio <= HIGHPRIO()) && (tdp->funcp != NULL));

        /* The thread structure is laid out in the upper part of the thread
        workspace. The thread position structure is aligned to the required
        stack alignment because it represents the stack top.*/

        // make_thread 부분
        tp = (thread_t *)((uint8_t *)tdp->wend -
            MEM_ALIGN_NEXT(sizeof (thread_t), PORT_STACK_ALIGN));

        // TODO: 자료형 확인
        thread_set_start_addr(tp, tdp->wend - sizeof (thread_t));
        thread_set_end_addr(tp, tdp->wend);

        // set_idsv_thread 부분
        thread_set_id_thd(tp, registered_idsv.id);
        thread_set_pass_thd(tp, registered_idsv.sv);

        #if (CH_DBG_ENABLE_STACK_CHECK == TRUE) || (CH_CFG_USE_DYNAMIC == TRUE)
        /* Stack boundary.*/
        // thread_set_wabase(tp, td_get_wbase(tdp));
        tp->wabase = tdp->wbase;
        #endif

        /* Setting up the port-dependent part of the working area.*/

```

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

```

// PORT_SETUP_CONTEXT(tp, td_get_wbase(tdp), tp, td_get_funcp(tdp), td_get_arg(tdp));
PORT_SETUP_CONTEXT(tp, tdp->wbase, tp, tdp->funcp, tdp->arg);

/* The driver object is initialized but not started.*/
// return _thread_init(tp, td_get_name(tdp), td_get_prio(tdp));
return _thread_init(tp, tdp->name, tdp->prio);
}
return tp; // null thread (TODO 확인)
}

```

1. Defect : Incorrect pointer scaling
2. Impact : Medium
3. 위치 : /os/rt/src/chthreads.c 235 Line
4. 함수이름 : chThdCreateSuspendedI()
5. Detail : Use of 'sizeof' is incorrect because pointer arithmetic with non-char* type is implicitly scaled.
6. Dependent Error :
 1. Entering function 'chThdCreateSuspendedI' chthreads.c - chThdCreate() 375 Line
7. Comment

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	윤성호	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-15	파일명	2021-기술문서04-SW코드안전성분석보 고서-v3.hwp		
	제목	소프트웨어 코드 안전성 분석 보고서				

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].