# 선박 기자재 탑재 SW 정보 수집 및 취약점 분석

실제 환경 기반의 분석을 통한 취약점 도출


| | | | |
|---|---|---|---|
| **제출일** | 2025. 08. 08 | **팀명** | SeaBugs |
| **담당멘토** | 이민우 | **담당PL** | 노용훈 |
| **팀원** | 이태용(PM), 유인서, 김주찬, 이규민, 윤현빈, 양승권, 진형권, 박지후 | | |

[목차]

# 1. 프로젝트 소개

## 1-1. 팀 소개

본 프로젝트는 SeaBugs 팀이 수행하였다. SeaBugs는 정보보안과 소프트웨어 취약점 분석에 관심이 있는 학생들이 모여, 해양 산업 내 선박 기자재의 소프트웨어 정보를 체계적으로 수집하고 잠재적 취약점을 분석하는 것을 목표로 활동한다.

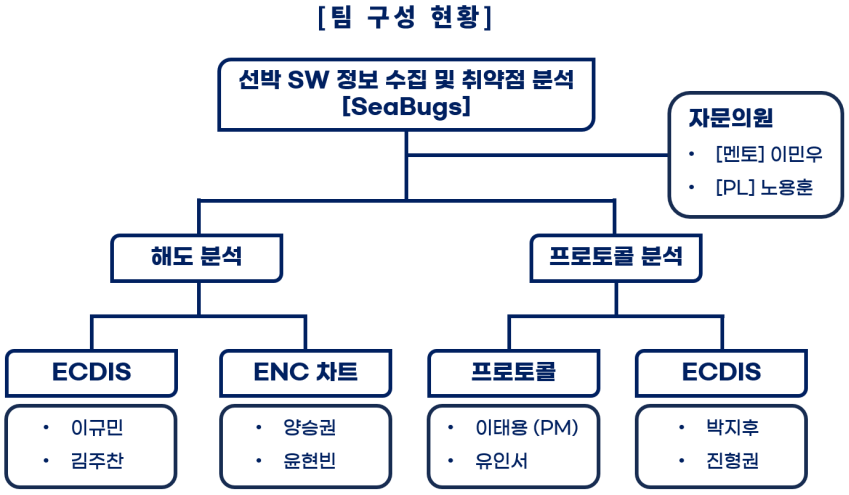



**그림 1 팀 구성도**

팀은 멘토와 PL을 제외한 8명으로 구성되었으며, 프로젝트 매니저(PM)를 중심으로 프로토콜 분석팀(4명)과 해도 분석팀(4명)으로 나누어 효율적인 분업 체계를 구축하였다. 모든 팀원이 소프트웨어 구조 분석과 취약점 분석 과정에 함께 참여하여, 선박 소프트웨어의 보안 위협을 다양한 관점에서 종합적으로 분석하였다.

## 1-2. 프로젝트 소개

### 1-2-1. 선행 연구 및 배경 지식

최근 해양 운송 산업은 디지털 전환이 가속화되면서 항해 및 통신 시스템이 대부분 소프트웨어(SW) 기반 전자 장비로 대체되고 있다. 대표적으로 ECDIS(Electronic Chart Display and Information System)와 AIS(Automatic Identification System)는 항로 계획, 통신, 충돌 방지 등 항해 필수 기능을 수행한다. 이러한 선박 SW는 항해 안전과 직결되며, 실시간 외부 데이터 수신을 기반으로 작동한다. 그러나 보안성이 충분히 고려되지 않은 폐쇄적 설계나, 오래된 통신 프로토콜(NMEA 0183/2000 등) 사용으로 인해 사이버 공격에 취약하다. 실제로 해양 사이버 보안 사고는 증가하고 있으며, AIS 스푸핑·GPS 신호 위조·전자해도 조작 등은 국제해사기구(IMO)에서도 중대한 위협으로 경고된다.

## 1-2-2. 프로젝트 목적 및 필요성

선박에 탑재되는 항해·통신용 소프트웨어는 위치 결정, 항로 계획, 주변 선박 식별, 해양 경고 수신 등 운항의 핵심 기능을 수행한다. 대표적으로 ECDIS와 AIS 시스템은 외부와 지속적으로 통신하며 데이터를 수신하기 때문에 사이버 위협에 노출될 가능성이 높다. 최근 해양 산업의 디지털화와 함께 이러한 보안 리스크는 현실적인 위협으로 부각되고 있다. 국제해사기구(IMO)는 2021년부터 사이버 보안 관리체계(Cyber Risk Management) 도입을 의무화하였으며, 선박 보안은 필수 요건이 되었다.

그러나 다음과 같은 문제가 여전히 존재한다.

1. 다수의 상용 선박 SW는 폐쇄적 구조로 인해 외부 보안 검토가 어렵다.
2. 사용되는 통신 프로토콜(NMEA 0183/2000)은 암호화·인증 기능이 없어 매우 취약하다.
3. 보안 패치가 느리거나 대응 체계가 부재한 경우가 많다.

이러한 문제는 AIS 스푸핑, GPS 신호 위조, 전자해도 조작 등 보안 사고로 이어질 수 있으며, 이는 선박 충돌, 항로 이탈, 해양 사고 등 물리적 재해로 확산될 가능성이 있다. 따라서 선박 SW에 대한 선제적 취약점 분석과 실증 기반 위협 평가는 해양 안전 확보와 국제 기준에 부합하는 보안 대응 체계 구축을 위한 중요한 기초 자료가 된다.

## 1-2-3. 선행 연구 및 배경 지식

선박 SW 개념 및 구성 요소

**1) ECDIS (Electronic Chart Display and Information System)**

ECDIS는 전자해도 기반의 항해 시스템으로, 선박의 위치와 항로를 디지털 해도에 표시하고 계획할 수 있도록 지원한다. 국제해사기구(IMO)는 대형 상선 및 여객선에 대해 ECDIS 장비의 의무 탑재를 규정하고 있으며, 종이 해도의 대체 수단으로 기능한다. 주요 기능은 다음과 같다.

- 선박 위치, 방향, 속도의 실시간 표시 (GPS, AIS 연동)
- 항로 계획 및 자동 항법 경로 안내
- 주변 장애물 경보 및 해도 기반 위험 감지
- 외부 센서 데이터(NMEA 메시지) 연동

본 프로젝트에서는 ECS 오픈소스 O프로그램을 주요 분석 대상으로 삼았다.

**2) NMEA 프로토콜 (0183/2000)**

NMEA(National Marine Electronics Association) 프로토콜은 선박 내 항해 장비 간 통신을 위한 표준 규격으로, 다음 두 가지 버전이 있다.

- NMEA 0183: ASCII 기반 직렬 통신 포맷. 구조가 단순하고 범용성이 높지만 보안 기능이 전무하다. 예: $GPRMC, $GPGGA, !AIVDM
- NMEA 2000: CAN 버스 기반 이진 포맷. 속도와 신뢰성이 높으나 폐쇄적이며 주로 상용 장비에서 사용된다.

본 프로젝트에서는 NMEA 0183 기반 메시지를 조작하여 AIS 데이터 위변조 및 파싱 테스트를 수행하였다.

**3) AIS 메시지 (Automatic Identification System)**

AIS는 선박 간 충돌 방지와 위치 공유를 위한 실시간 위치 교환 시스템이다. 각 선박은 MMSI, 위치, 속도, 진행 방향(COG), 선박 타입 등의 정보를 지속적으로 전송하며, 주변 선박의 AIS 신호를 수신해 시각화한다. AIS 메시지는 대부분 NMEA 0183 포맷의 !AIVDM 문자열로 전송되며 다음과 같은 특성을 가진다.

- 메시지가 공개되어 있어 쉽게 생성·조작 가능
- CRC 오류 검출 외 인증 절차 부재
- 수신 측 파서에 따라 파싱 오류·버퍼 오버플로우 발생 가능

본 프로젝트에서는 AIS 메시지를 생성·변형하여 오픈소스 O프로그램에서 발생할 수 있는 취약점을 실험적으로 분석하였다.

이러한 요소들은 모두 선박 SW의 핵심 구성 요소이자 잠재적 보안 취약점이 발생할 수 있는 주요 지점이다. 본 프로젝트는 이들의 구조와 동작 방식을 기반으로 실험 환경을 구축하고 다양한 보안 테스트를 수행하였다.

.

# 2. 수행결과

## 2-1. WBS 달성 결과

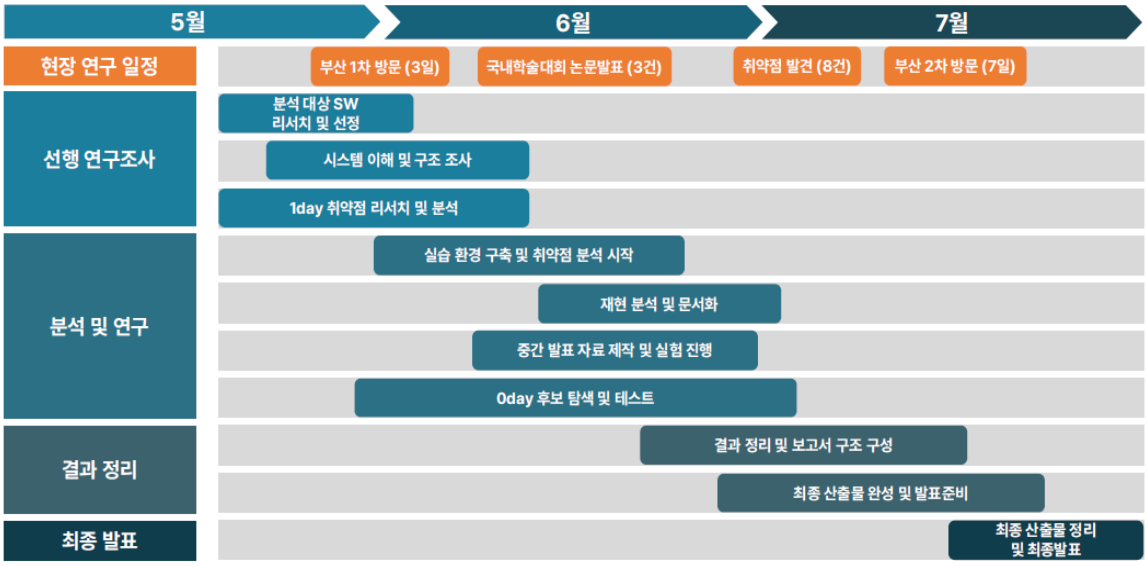



**그림 2 프로젝트 추진 일정(WBS)**

## 2-2. 프로젝트 수행 환경

### 1) 협업 도구

- Notion: 프로젝트 문서화 및 일정 관리 등 전반적인 자료 정리를 위한 플랫폼
- GitHub: 코드 및 분석 자료의 버전 관리를 위한 저장소
- 카카오톡: 팀원 간의 상시 소통을 위한 메신저
- Discord, Google Meet: 온라인 회의 진행 시 주로 활용한 플랫폼
- VirtualBox: 테스트 및 분석을 위한 가상 머신 환경 프로그램

### 2) 프로젝트 진행 방식

- 공지 방식: 카카오톡으로 공지하며, 이모지를 통한 확인 응답
- 자료 공유: 정기 회의 전일까지 Notion에 업로드
- 실험 기록: 실패 사례도 반드시 기록하여 팀원들의 학습 자료로 활용

### 3) 의사 결정 방식

- 소규모 작업 방향: 팀원 3인 이상 동의 시 즉시 결정

- 주요 변경 사항 (예: 분석 대상 변경, 실험 방식 조정 등): 정기 회의에서 과반수 동의 후 결정

#### 4) 회의 운영

- 정기 회의: 주 1회 오프라인 회의 진행 (매주 일요일, 사전에 지정된 장소)
  (참석 불가 시, PM과 협의 및 불참 사유서 작성 후 온라인 참여)
- 주중 자율 회의: 가능한 인원이 자율적으로 온라인(디스코드) 회의 진행

## 2-3. 주요 성과

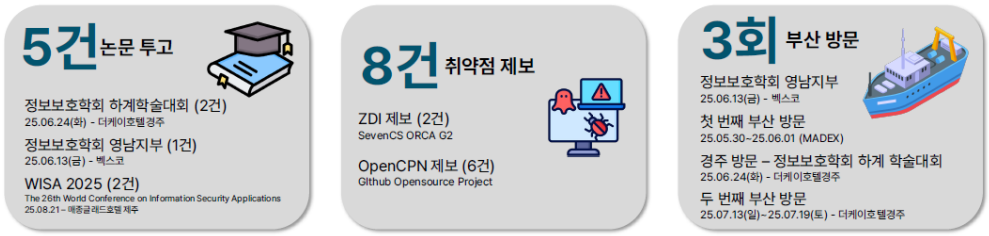



**그림 3 프로젝트 산출물**

#### 1) 실험 환경 및 프로토콜 테스트 구성

**그림 4 Virtual Sailor NG ↔ O프로그램**

- Virtual Sailor NG와 오픈소스 O프로그램 간 COM 포트 연결
- Virtual Sailor NG에서 NMEA 신호 전송, O프로그램에서 수신·시각화 확인

### 2) 취약점 8건 도출

| NO. | 취약점 제목 | 분류 | 프로그램 | 제보날짜 |
|---|---|---|---|---|
| 1 | OpenCPN – Symbol Manipulate (Integrity Bypass) | Integrity Bypass | OpenCPN | 25.07.01 |
| 2 | OpenCPN – File overwrite (Path traversal) | Path Traversal | OpenCPN | 25.07.10 |
| 3 | OpenCPN – Code Injection (RCE) | RCE | OpenCPN | 25.07.10 |
| 4 | OpenCPN - Chart Downloader Plugin (RCE) | RCE | OpenCPN | 25.07.15 |
| 5 | OpenCPN – Command Injection (RCE) | RCE | OpenCPN | 25.07.25 |
| 6 | OpenCPN – heap Overflow (System Dos) | Buffer Overflow | OpenCPN | 25.07.24 |
| 7 | ORCA G2 – TOCTOU (LPE) | Race Condition, LPE | SevenCs Orca G2 | 25.07.16 |
| 8 | ORCA G2 - Unrestricted \\.\C: Access (System Dos, LPE) | System DoS, LPE | SevenCs Orca G2 | 25.07.26 |


**그림 5 취약점 제보 내역**

- 상용 S사 O프로그램 취약점 2건 → ZDI(Zero Day Initiative) 제보 완료
- 오픈소스 O프로그램 취약점 6건 → MITRE 제보 완료

### 3) 논문 5편 투고 완료

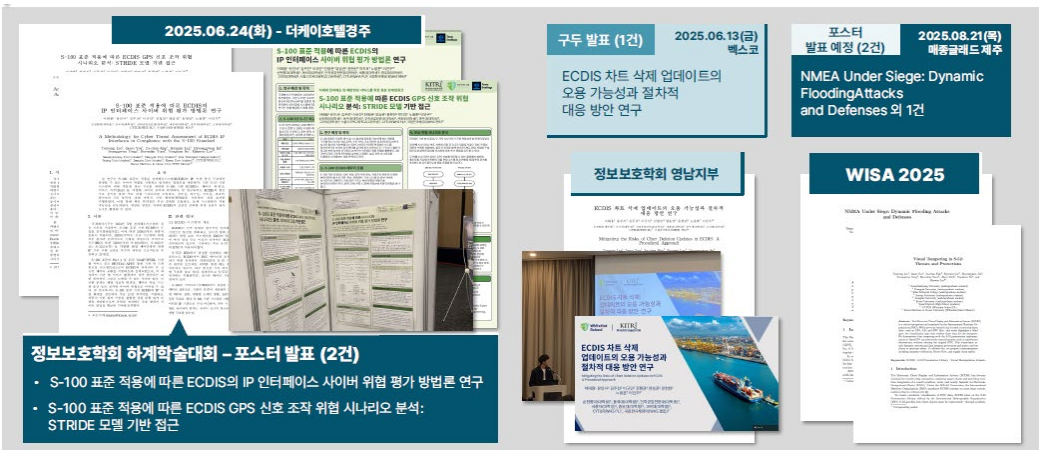



**그림 6 논문 5편 투고**

- 정보보호학회 영남지부 학술대회 1편(구두 발표) – 2025.06.13
- 정보보호학회 하계 학술대회 2편(포스터 발표) – 2025.06.24
- WISA 2025 2편(포스터 발표) – 2025.08.21

**4) 국내 유관 기관 다수 방문**

- 동남 정보보호 클러스터, 부산항 VTS(해상교통관제센터) 방문
- 한국해양대학교 방문
  - 융합보안연구실
  - 실습선(한나라호)
  - 전기추진체계실험실
  - GMDSS 실습실

# 3. 수행결과 상세소개

## 3-1. 주차 별 분석 현황

1주차

- 프로젝트 그라운드룰 협의 및 사전 회의
- 선박 관련 용어 탐구 및 분석 대상 선정
- 가상머신(VM) 구성, 프로그램 설치 및 구조 분석

2주차

- 팀 구성(해도 분석팀, 프로토콜 분석팀)
- 분석 결과를 바탕으로 논문 작성 착수(정보보호학회 하계학술대회)
- 대상 프로그램 정적 분석 및 기능 파악

3주차

- Virtual Sailor를 이용한 프로토콜 테스트 환경 구성
- 해도 파싱 모듈 상세 분석
- 동적 분석과 정적 분석을 병행하여 프로그램 흐름 파악

4주차

- 해도 심볼 위변조 분석 및 실험
- S-57, S-63 규격 문서화

- NMEA·AIS 프로토콜 구조 분석 및 송수신 테스트

5주차

- XML 파일 분석 및 퍼징 시도
- 파싱 결과를 토대로 추가 분석 지점 도출 및 향후 분석 방향 설정

6주차

- 한국정보보호학회 영남지부 논문 포스터 게시
  (S-100 기반 ECDIS GPS 신호 조작 위협 시나리오 분석: STRIDE 모델 기반 접근, S-100 기반 ECDIS의 IP 인터페이스 사이버 위협 평가 방법론 연구)

7주차

- chartsymbols.xml 파일 조작 실험 → Chart Downloader Plugin RCE 취약점 보고로 연계

8주차

- 한국정보보호학회 하계학술대회 논문 발표 및 포스터 게시
  (ECDIS 차트 삭제 업데이트의 오용 가능성과 절차적 대응 방안 연구)
- 오픈소스 O프로그램 AIS 위조 신호 테스트베드 구현 → AIS 동적 플러딩 시나리오 논문
  (NMEA Under Siege: Dynamic Flooding Attacks and Defenses) 로 연계

9주차

- Path Traversal 취약점 발견 및 보고서 작성
- 상용 SW 정적 분석 착수
- AIS 동적 플러딩 시나리오 논문 작성

10주차

- 오픈소스 SW 환경에 AddressSanitizer 적용 및 빌드
- 메모리 오류 탐지 및 취약 동작 포착 시도(heap overflow 등)
- Valgrind 기반 메모리 누수 분석 환경 구축
- 상용 SW 동적 분석 시작

11주차

- 분석 주제 통합 회의 및 최종 보고서 목차 논의
- 부산 오프라인 연구 모임 계획 수립

12주차

- GMDSS 및 한나라호 체험을 통한 실제 해양 SW 운용 환경 확인
- PoC 기반 취약점 리포트 작성 착수
- 발표 자료, 최종 보고서 및 산출물 정리 본격화

마무리

- 전체 분석 결과 정리 및 산출물 통합
- 도출된 취약점을 기반으로 논문 제작 방향 논의
- 향후 발표 및 외부 보고용 콘텐츠 제작 준비

## 3-2. 취약점 분석[1]

### 3-2-1. 경로 검증 부재로 인한 Path Traversal

코드 검토 과정에서, 특정 플러그인 로드 기능이 외부 입력값으로부터 전달되는 파일 경로를 별도의 검증 없이 처리하는 동작을 확인하였다. 이 로직은 디렉터리 이동(../)과 같은 경로 조작 문자를 포함한 입력을 허용하므로, 지정된 작업 디렉터리를 벗어난 임의 위치의 파일에 접근하거나 덮어쓰기가 가능하다.

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin version="1">
  <name>../../../hijack</name>
  <version>0.0.1</version>
  <release>0</release>
  <summary>PoC</summary>
  <description>Path Traversal PoC</description>
  <target>ubuntu-x86_64</target>
  <build-target>ubuntu</build-target>
  <build-gtk>gtk3</build-gtk>
  <target-version>24.04</target-version>
  <target-arch>x86_64</target-arch>
  <api-version>1.18</api-version>
  <tarball-url>file:///nope</tarball-url>
</plugin>
```

**[그림 1] Path Traversal 유발 문자열을 포함한 XML PoC 예시**

[1] 본 보고서에 기술된 모든 취약점은 실제 장비가 아닌 테스트용 가상 환경에서 재현 및 PoC를 수행하였다.

```
user@KARAJAN-TEST:~/OpenCPN/build_default/cli$ ./opencpn-cmd import-plugin ./poc/poc.tar.gz
user@KARAJAN-TEST:~/OpenCPN/build_default/cli$ cd ~
user@KARAJAN-TEST:~$ ls
OpenCPN  hijack.dirs  hijack.files  hijack.version
user@KARAJAN-TEST:~$
```

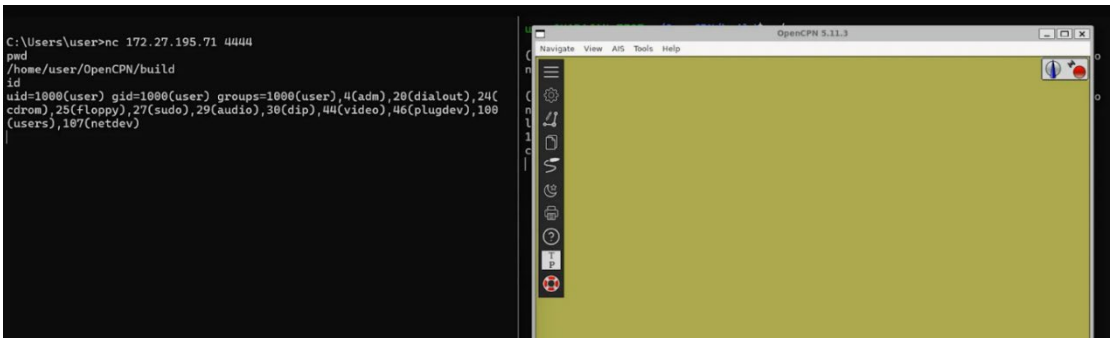
**[그림 2] 테스트 환경에서 PoC 실행 결과**

이 취약점은 중요 설정 파일 변조, 악성 코드 설치, 로그 조작 등으로 확장될 수 있으며, 시스템 안정성과 무결성에 심각한 영향을 줄 수 있다.

### 3-2-2. 플러그인 로드 시 검증 불충분으로 인한 코드 실행

코드 검토 과정에서, 플러그인 로드 기능이 외부 모듈을 불러올 때 화이트리스트 기반의 신뢰성 검증 없이, 단순 블랙리스트 방식의 제한만 적용하는 동작을 확인하였다. 이로 인해 공격자는 정상 구조를 가진 악성 모듈을 제작하여 프로그램에 로드시킬 수 있으며, 이는 임의 코드 실행으로 이어질 수 있다.

PoC에서는 테스트용 플러그인 템플릿에 명령 실행 코드를 삽입하여 빌드한 뒤, 플러그인 로드 기능을 통해 실행하였다.




**[그림 3, 4] PoC 실행 결과 – 임의 명령 실행 성공**

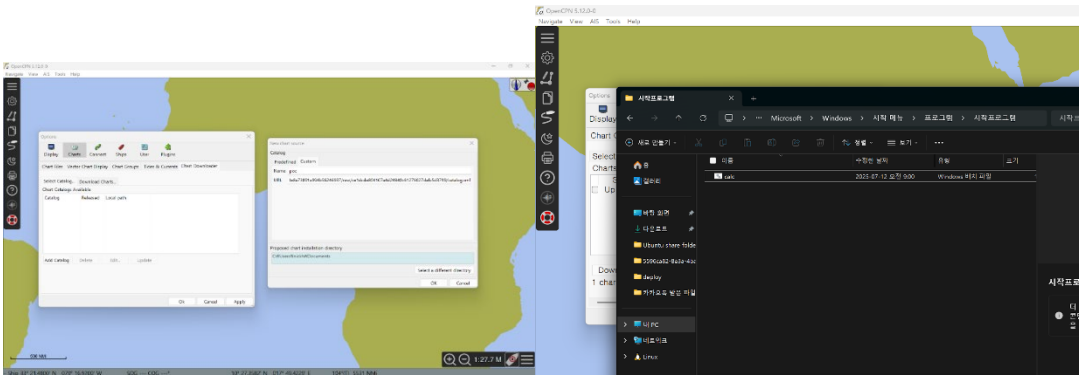
이 취약점은 원격 코드 실행(RCE)이 가능하므로, 시스템 제어권 탈취, 데이터 유출, 서비스 중단 등 심각한 보안 위협으로 이어질 수 있다

### 3-2-3. 파일 경로 검증 부재로 인한 원격 코드 실행 (RCE)

특정 차트 다운로드 기능이 외부에서 가져온 파일을 적절한 검증 없이 사용자가 지정한 로컬 디렉터리에 해제하는 동작을 확인하였다.

이 과정에서 경로 조작(../)을 이용하면 지정된 디렉터리를 벗어나 임의 경로에 파일을 생성하거나 덮어쓸 수 있다.

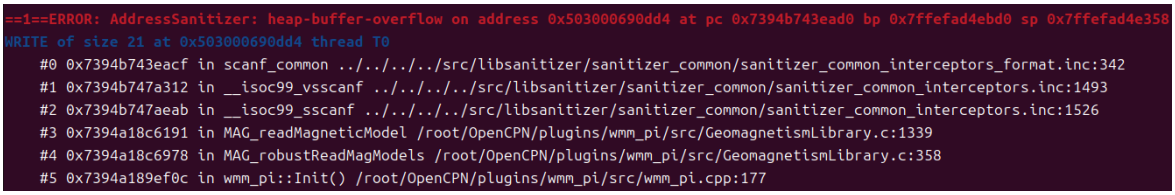
PoC에서는 조작된 차트 카탈로그 XML 내부에 시작 프로그램 경로를 포함시켜, 상위 디렉터리로 탈출한 뒤 악성 BAT 파일을 생성하였다. 이 파일은 사용자의 시작 프로그램 폴더에 위치하게 되며, 시스템 재부팅 시 자동으로 실행된다.



**[그림 4, 5] 차트를 다운 받고 시작프로그램에 악성프로그램이 저장된 결과**

이 취약점은 물리적 접근 없이도 원격에서 악성 코드를 설치·실행할 수 있어, 시스템 장악, 지속적 침투, 추가 공격 수행 등 심각한 위협을 초래할 수 있다.

### 3-2-4. 입력 길이 검증 미흡으로 인한 Heap Buffer Overflow




**[그림 5] 테스트 환경에서 AddressSanitizer가 탐지한 힙 버퍼 오버플로우 오류**

AddressSanitizer 빌드 환경에서 대상 프로그램을 실행한 결과, 특정 파일을 파싱하는 과정에서 크래시가 발생하였다. 분석 로그에 따르면, 힙 버퍼 경계를 초과하는 쓰기(write) 동작이 감지되었다. 이는 입력 길이 초과 또는 포맷 파싱 로직의 문제 가능성을 시사한다.

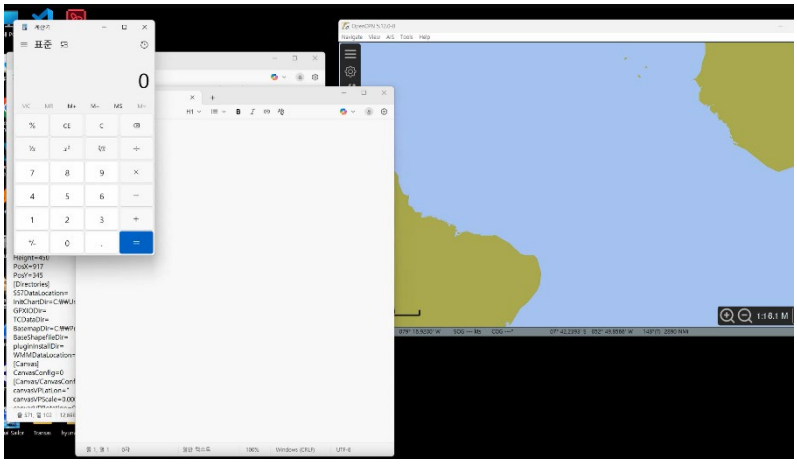
추가 분석 결과, 문자열 입력을 처리하는 함수 호출에서 고정 길이 제한 값이 선언되어 있으나, 널 종단 문자까지 포함해 실제로는 더 많은 바이트가 메모리에 기록되는 동작이 확인되었다. 입력 값이 정상적으로 잘리지 않거나 포맷 제약을 우회하는 경우, 예상보다 많은 데이터가 복사되어 인접 힙 메모리 영역을 침범하게 된다.

힙 기반 버퍼 오버플로우가 발생하는 조건은 다음과 같다.

- 파일 구조 내 특정 필드에 제한 길이를 초과하는 문자열이 포함된 경우
- 해당 필드를 처리하는 과정에서 추가적인 길이 검증 없이 복사가 이루어지는 경우
- 고정 크기보다 작은 메모리 버퍼가 할당되어 인접 메모리에 대한 비의도적 쓰기가 발생하는 경우

이러한 구조적 결함은 악의적으로 조작된 입력 파일을 통해 힙 영역이 손상될 수 있으며, 서비스 거부(DoS)뿐 아니라 임의 코드 실행 가능성으로 이어질 수 있는 심각한 보안 취약점이다.

### 3-2-5. 명령 실행 기능의 입력 검증 미흡으로 인한 Command Injection

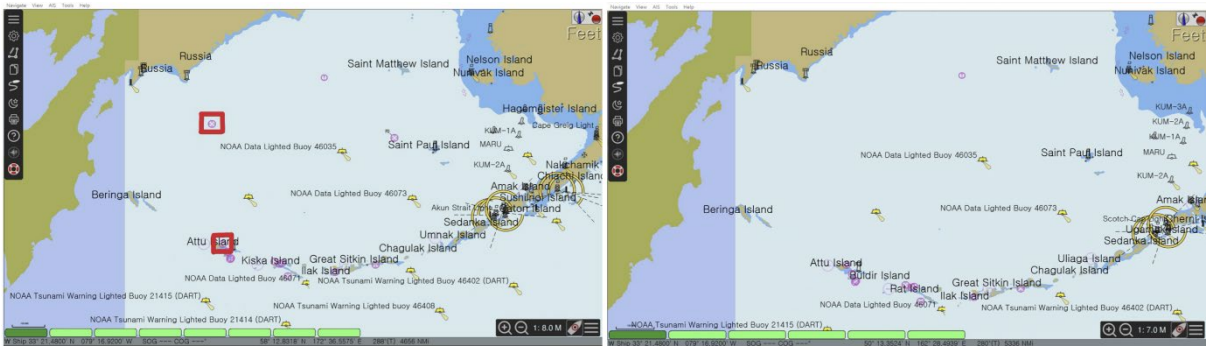


**[그림 6] 테스트 환경에서 명령어 삽입 성공 후 계산기 실행**

해당 플러그인은 프로그램 내 버튼을 통해 외부 프로그램이나 명령어를 실행할 수 있도록 설계되어 있다. 그러나 사용자로부터 입력받은 명령 문자열을 적절히 필터링하거나 이스케이프 처리하지 않고 운영체제 쉘에 직접 전달하는 로직이 확인되었다.

이로 인해 공격자는 &, | 등의 쉘 메타문자를 이용해 원래 의도와 무관한 임의의 명령어를 실행할 수 있다. 해당 취약점은 원격 코드 실행(RCE)로 이어질 수 있으며, 시스템 장악·데이터 유출·서비스 마비 등 심각한 피해를 유발할 수 있다.

### 3-2-6. 시각 정보 조작을 통한 장애물 은폐

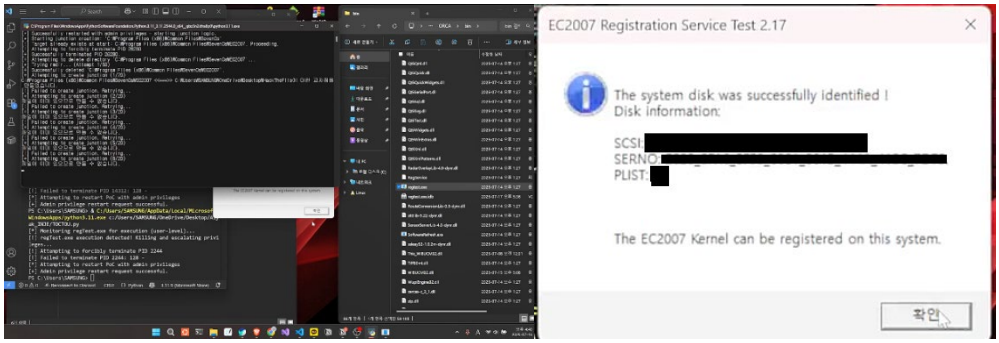



**[그림 7] 화면 변조 전후 비교 (좌: 변조 전, 우: 변조 후)**

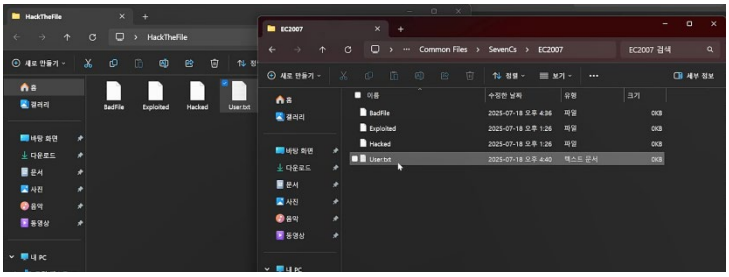
공격자는 특정 시각화 규칙 파일을 복사한 뒤, 위험 표식을 정의하는 항목을 제거 또는 주석 처리한 수정본을 생성하고 원본 파일을 교체할 수 있다. 대상 프로그램은 파일 내용의 무결성을 검증하지 않고, 이름만 일치하면 그대로 로드하여 시각화에 사용한다.

이로 인해 해도 렌더링 과정에서 수중 장애물과 같은 위험 표식이 표시되지 않게 되며, 항해자가 이를 인지하지 못할 경우 안전 운항에 심각한 위협이 될 수 있다.

### 3-2-7. 디렉터리 생성·복사 과정의 경로 검증 미흡으로 TOCTOU기반 권한 상승




**[그림8, 9] 정상 실행 화면**




**[그림9] Junction 디렉터리 삽입**

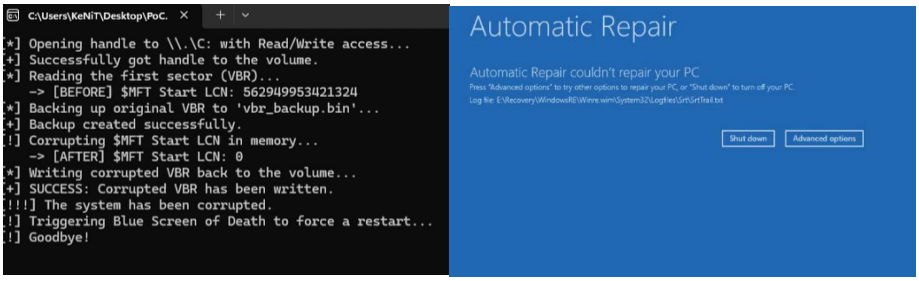
해당 취약점은 특정 프로그램의 디렉터리 생성 및 파일 복사 과정에서 발생한다. 프로그램은 대상 경로의 존재 여부를 확인한 후, 존재하지 않으면 디렉터리를 생성하고 파일을 복사하는 절차를 수행한다.

그러나 이 과정에서 각 단계 사이에 경로의 무결성을 재검증하지 않으므로, 공격자가 경로를 삭제한 뒤 심볼릭 링크(Junction)로 외부 경로를 지정할 수 있다. 이로 인해 원래 의도된 위치가 아닌 관리자 전용 디렉터리에 파일을 생성·변경하거나, 임의의 실행 파일을 덮어쓸 수 있다.

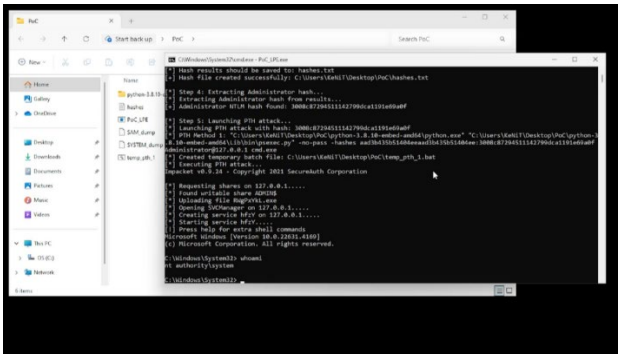
이러한 구조는 전형적인 TOCTOU(Time-of-Check to Time-of-Use) 취약점이며, 로컬 권한 상승(LPE), 중요 파일 무결성 훼손, 임의 코드 실행 등의 보안 위협으로 이어질 수 있다.

### 3-2-8. 저장 장치 접근 권한 설정 취약점에 따른 System DoS 및 SAM Hive 덤프

해당 취약점은 라이선스 인증 확인을 수행하는 서비스 구성 요소에서 발생한다.
서비스는 하드웨어 정보를 기반으로 고유 식별값(System ID)을 생성하는 과정에서 로컬 디스크 장치에 직접 접근한다. 이때 디스크 장치에 대한 접근 권한이 모든 사용자(EVERYONE) 그룹에 읽기/쓰기(RW)로 부여되어 있어, 비관리자 계정도 원시 디스크에 직접 읽기·쓰기 작업을 수행할 수 있다.




**[그림10, 11] 테스트 환경에서 부팅 불능 상태에 빠진 장면**



**[그림12] 테스트 환경에서 SAM/SYSTEM 덤프를 통한 권한 상승 시도**

**1) System DoS**

공격자는 원시 디스크의 부트 섹터(VBR) 일부를 변조하여 파일 시스템 메타데이터($MFT)의 시작 위치를 손상시킬 수 있다. 이로 인해 운영체제가 정상적으로 부팅되지 않아 시스템이 마비된다.

**2) Local Privilege Escalation (LPE)**

동일한 권한 설정 문제를 악용하면, 공격자는 시스템 구성 파일(SAM 및 SYSTEM 하이브)을 원시 디스크에서 직접 읽어낼 수 있다. 이 데이터로부터 계정 해시를 추출하고, 해당 해시를 활용한 Pass-the-Hash(PtH) 공격을 통해 관리자 권한의 셸을 획득할 수 있다.

이 취약점은 단일 권한 설정 오류가 서비스 거부(DoS)와 로컬 권한 상승이라는 두 가지 심각한 보안 위협을 모두 야기할 수 있다는 점에서 치명적이다.

## 3-3. 취약점을 활용한 선박 사이버 공격 시나리오 (ECDIS 무력화)


**그림 7 취약점을 활용한 선박 사이버 공격 시나리오**

본 시나리오는 특정 항해 보조 소프트웨어의 취약점을 악용하여 ECDIS(전자해도표시시스템)를 무력화하는 공격 과정을 단계별로 나타낸 것이다. 공격자는 해운사·항해사와의 통신 환경 및 시스템 구조를 사전에 분석한 뒤, 악성 전자해도(S-100 형식)를 제작·전달하여 항해 보조 시스템을 마비시킨다. 공격 단계는 다음과 같다

1. **정보 수집**
   해운사 및 선박 환경에 대한 사전 정찰을 수행하고, 항해사·선박 운영자 대상 스피어 피싱에 필요한 정보를 수집한다.

2. **악성 해도 제작**
   신규 전자해도(S-100 형식)에 악성 코드(트로이 목마)를 삽입하여 제작한다.

3. **스피어 피싱 메일 발송**
   선원 위성 이메일 서비스를 이용해 악성 전자해도를 첨부한 스피어 피싱 메일을 항해사에게 발송한다.

4. **악성 코드 실행**
   항해사가 첨부 파일을 열면, 저장 장치를 통해 선내 폐쇄망에 악성 코드가 주입·실행된다.

5. **시스템 무력화**
   악성 코드가 ECDIS 운영 환경을 손상시켜 항해 보조 시스템이 정상적으로 작동하지 않게 하며, 결과적으로 항로 계획·위험 회피 기능이 마비된다.

## 3-4. 논문 소개

### 3-4-1. ECDIS 차트 삭제 업데이트의 오용 가능성과 절차적 대응 방안 연구 (정보보호학회 영남지부, 2025.06.13)

ECDIS 시스템의 Edition 0 업데이트는 차트 객체 삭제를 의미하며, 사용자의 부주의 또는 악의적 조작으로 항해 필수 객체가 사라질 수 있는 취약성을 내포한다. 이 연구는 해당 절차의 실제 운용 흐름을 분석하고, 차트 삭제 업데이트의 오용 가능성을 시나리오 기반으로 도출하였다. 이를 토대로 절차적 오남용을 예방할 수 있는 5단계 Safe-Delete SOP를 제안하였으며, 시스템을 수정하지 않고도 즉시 적용 가능한 실무 대응 방안임을 입증하였다. 본 연구는 차트 데이터 관리 절차의 안전성을 강화하고, 항해 안전 확보를 위한 표준 운영 지침 마련에 기여한다.

### 3-4-2. S-100 표준 적용에 따른 ECDIS GPS 신호 조작 위협 시나리오 분석: STRIDE 모델기반 접근 (정보보호학회 하계학술대회, 2025.06.24)

S-100 기반 ECDIS 시스템은 XML과 REST API를 통한 실시간 데이터 연동 기능을 제공하지만, 개방형 구조 특성상 사이버 위협에 취약하다. 본 연구에서는 STRIDE 모델을 활용하여 ECDIS의 취약 지점을 체계적으로 분석하고, ENC 파일 변조와 GPS 신호 조작이라는 두 가지 핵심 위협 시나리오를 도출하였다.

이들 시나리오는 악의적 조작 시 항로 오류, 좌초, 충돌과 같은 심각한 항해 사고를 유발할 수 있으며, 공격 트리를 통해 구체적인 공격 경로와 파급 효과를 시각화하였다. 연구 결과를 바탕으로, 시스템 수정 없이도 적용 가능한 대응 방안으로 TLS 기반 암호화, 디지털 서명 검증, XML 데이터 필터링을 제시하였다. 또한 향후에는 머신러닝 기반 이상 탐지 및 국제 협력 보안 프레임워크 수립이 필요함을 강조하였다. 본 연구는 표준 보안 가이드라인이 부재한 상황에서 실무 대응의 타당성을 확보하고, S-100 기반 ECDIS의 사이버 보안 강화를 위한 기초 자료로 활용 가능함을 입증하였다.

### 3-4-3. S-100 표준 적용에 따른 ECDIS의 IP 인터페이스 사이버 위협 평가 방법론 연구 (정보보호학회 하계학술대회, 2025.06.24)

본 연구는 S-100 표준이 적용된 차세대 전자해도시스템(ECDIS)의 IP 인터페이스 사이버 위협을 평가하였다. 국제해사기구(IMO)는 2024년 5월 S-100 기반 ECDIS 도입을 공식화하였으며, 2026년부터 자발적 운용이 허용되고 2029년부터 의무 탑재가 시행될 예정이다.

기존 S-57 기반 ECDIS가 오프라인 매체에 의존하는 폐쇄형 구조였던 것과 달리, S-100 기반 시스템은 IP 기반 실시간 데이터 교환을 적극 활용하는 개방형 구조를 가진다.
이러한 변화는 해양 정보 위변조, 데이터 유실, 시스템 중단 등 심각한 사이버 위협에 노출될 가능성을 높인다. 연구에서는 ECDIS의 통신 구조를 분석하여 주요 위협 시나리오를 도출하고, 침투성, 복구성, 지속성, 파급력, 탐지성의 5개 항목에 대해 가중치 기반 합산법(WSM)을 적용하여 정량적으로 위험도를 평가하였다. 그 결과, 기상 데이터 위조 시나리오가 총점 13점으로 가장 높은 위험도를 기록하였다.

### 3-4-4. Visual Tampering in S-52: Threats and Protections
(WISA 2025, 2025.08.21)

전자해도표시 및 정보시스템(ECDIS)은 국제해사기구(IMO)가 선박 운용을 위해 의무화한 항해 안전 장비이다. 기존 ECDIS 사이버 보안 연구는 주로 GPS 스푸핑, AIS 신호 조작 등 외부 입력 데이터 무결성에 초점을 맞추어 왔다. 그러나 입력 데이터가 온전하더라도, 최종 시각 출력을 생성하는 S-52 라이브러리 규칙 구현 로직이 변조되면 항해자가 왜곡된 정보를 인지하지 못한 채 운항할 수 있으며, 이는 심각한 해양 사고로 이어질 수 있다.
본 연구에서는 오픈 소스 기반 O프로그램으로 가상 테스트베드를 구축하고, S-52 라이브러리 규칙 중 심볼 구성을 담당하는 chartsymbols.xml을 조작하여 암석(OBSTRN)과 같은 위험 표식을 안전 수심으로 위장하는 공격을 수행하였다.
실험 결과, 시각 출력 단계에서의 변조가 항해 안전에 직접적인 위협이 될 수 있음을 실증하였으며, 이를 통해 데이터 수신 단계에 편중된 기존 ECDIS 보안 연구의 범위를 시각화 단계까지 확장하는 필요성을 제시하였다.

### 3-4-5. NMEA Under Siege: Dynamic Flooding Attacks and Defenses
(WISA 2025, 2025.08.21)

NMEA(Networked Marine Electronics Association) 프로토콜은 선박 간 항해 정보를 교환하는 데 널리 사용되지만, 인증·암호화·무결성 검증 기능이 부재하여 보안에 취약하다. 이러한 구조적 한계로 인해 AIS(Automatic Identification System)를 통한 위치 정보가 대량 위조·조작될 경우, 운항자의 해상 상황 인식에 심각한 혼란을 초래할 수 있다.
본 연구에서는 오픈 소스 기반 O프로그램을 활용해 가상환경 테스트베드를 구축하고, NMEA 프로토콜 형식에 따른 AIS 신호 동적 플러딩 공격을 수행하였다. 실험 결과, 다수의 가짜 오브젝트가 생성되어 화면상에 시각적 혼선을 유도하는 상황을 관측하였으며, 이를 통해 NMEA 프로토콜의 보안 취약성을 실증하였다.
이 연구는 NMEA 프로토콜 보안 강화를 위한 추가 연구 및 표준 개선의 필요성을 강조하며, 해상 통신 프로토콜의 안전성을 높이기 위한 기초 자료로 활용될 수 있다.

## 3-5. 외부 활동 내역

### 3-5-1. 정보보호학회 영남지부

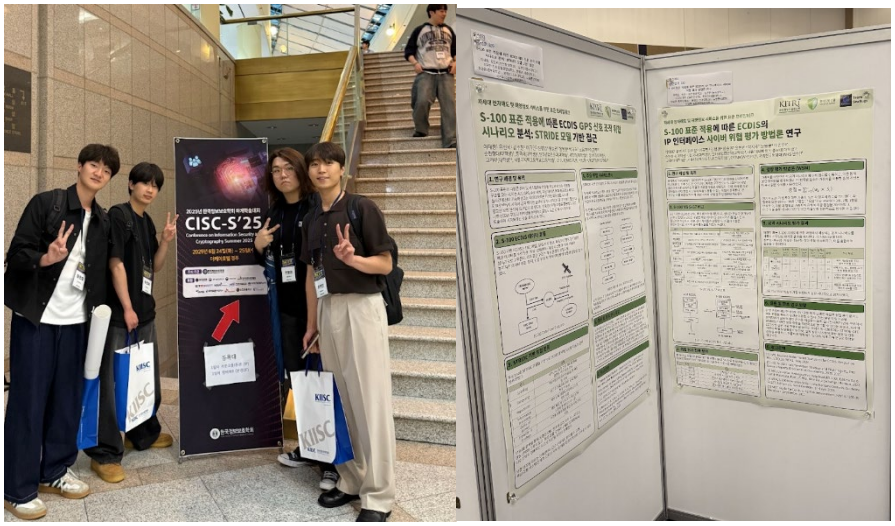

**[사진 1] 정보보호학회 영남지부 발표 장면**

활동 시기/장소: 2025.06.13(금), 벡스코

활동개요: 정보보호학회 영남지부 학술대회에 참석하여 "ECDIS 차트 삭제 업데이트의 오용 가능성과 절차적 대응 방안" 논문을 발표하였다.

성과: 연구 논문 구두 발표를 통해 차트 삭제 업데이트 절차의 보안 취약성과 대응 방안을 학계에 소개하였으며, 타 학회 참가자들과 해양 보안 주제를 논의하였다.

### 3-5-2. 정보보호학회 하계학술대회




**[사진 2,3] 정보보호학회 하계학술대회 포스터 발표 현장**

활동 시기/장소: 2025.06.24(화), 더케이호텔 경주

활동 개요: 국내 주요 정보보호 연구자와 산업 관계자가 참여한 하계 학술대회에서 팀 프로젝트 성과를 공유하였다. "S-100 표준 적용에 따른 ECDIS GPS 신호 조작 위협 시나리오 분석: STRIDE 모델 기반 접근" 및 "S-100 표준 적용에 따른 ECDIS의 IP 인터페이스 사이버 위협 평가 방법론 연구" 논문을 포스터 세션에서 발표하였다.

성과: S-100 기반 ECDIS의 구조적 취약성과 위협 시나리오를 체계적으로 제시하여 학계의 관심을 유도하였으며, 다른 연구팀과 연구 성과를 비교·분석하는 시간을 가졌다.

### 3-5-3. 동남 정보보호 클러스터


**[사진 4] 동남 정보보호클러스터 방문**

활동 시기/장소: 2025.05.30, 동남권 정보보호클러스터

활동 개요: 한국인터넷진흥원(KISA)에서 동남지역 특성에 맞춰 설립한 정보보호 클러스터를 방문하였다. 해당 클러스터는 전국에서 유일하게 선박 시스템 테스트베드를 보유하고 있으며, 해양 보안 및 산업 보안 관련 기업과 전문가들이 참여하는 기술 교류의 장으로 운영된다.

성과: 선박 시스템 테스트베드의 구성과 활용 사례를 직접 확인하고, 산업계의 실질적인 보안 수요와 해양 소프트웨어의 실제 적용 환경에 대한 통찰을 확보하였다.

### 3-5-4. 부산항 VTS


**[사진 5] 부산항 VTS 방문**

활동 시기/ 장소: 2025.05.30, 부산항 해상교통 관제센터(VTS)

활동 개요: 부산항 VTS의 역할과 지리적 위치에 대한 설명을 듣고, 해양 사이버 보안 측면에서 VTS 시스템이 갖는 중요성을 확인하였다. 관제실 내부를 시찰하며 AIS·레이더, ENC 등 주요 장비의 운용 현황을 직접 관찰하였다.

성과: VTS가 항만 안전과 해상 교통 관리에서 수행하는 핵심 역할을 이해하고, VTS 시스템 보안이 해양 사이버 보안 전반에서 차지하는 중요성을 재인식하였다.

### 3-5-5. 한국 선급 (KR) 수석님 미팅

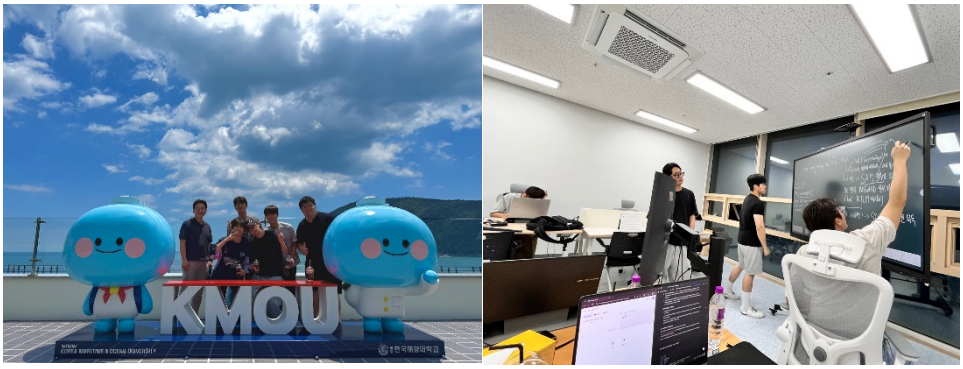
**[사진 6] 한국선급(KR) 임정규 수석님 미팅**

활동 시기/장소: 2025.05.31, 부산역 세미나실

활동 개요: 한국 선급 임정규 수석 검사원님과 만나 선박 보안에 대한 인식을 교류하고, 사이버 보안 요건 및 인증 절차와 관련한 의견을 나누었다. 또한 화이트햇스쿨 최초로 팀 프로젝트 세미나를 진행하였다.

성과: 실제 선급 심사 기준과 대응 방안을 공유받아 프로젝트 결과물의 현실성을 검토하였으며, 업계 전문가와의 네트워킹을 통해 향후 연구 방향에 대한 인사이트를 확보하였다.

### 3-5-6. 한국해양대학교 융합보안지식 연구실




**[사진 7,8] 해양대학교 단체사진 및 연구실**

활동 시기/장소: 2025.07.13~2025.07.19, 한국해양대학교 융합보안지식 연구실

활동 개요: 7일간 해당 공간을 프로젝트 분석 거점으로 활용하며, 팀원 간 실시간 피드백을 통해 체계적인 시나리오 분석을 진행하였다.

성과: 실제 ECDIS 실습 장비 구성을 확인하고, 집중 분석 기간 동안 실전 테스트 아이디어를 도출하였다. 또한 공동 작업 환경에서 팀워크와 분석 효율성을 향상시켰다.

3-5-7. 한국해양대학교 한나라호 실습실 (브릿지, 기관실)

**[사진 9,10] 해양대학교 한나라호 기관실 및 브릿지**

활동 시기/장소: 2025.07.13~2025.07.19, 한국해양대학교 실습선 한나라호

활동 개요: 실제 항해 중인 한나라호에 탑승하여 브릿지와 기관실에서 ECDIS, GMDSS, 레이더 등 주요 항해 장비의 운용을 실습하였다.

성과: 실전 환경에서 선박 소프트웨어의 운용 흐름을 이해하고, 해양 장비 간 물리적 연결 구조를 학습하였다. 이를 통해 향후 보안 분석 시 하드웨어·소프트웨어 연계 구조를 고려한 시나리오 설계에 활용 가능성을 확보하였다.

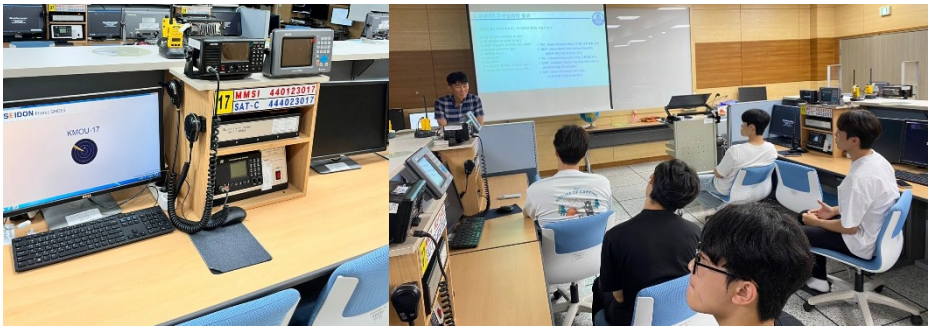
3-5-8. 한국해양대학교 전기추진체계실험실

활동 시기/장소: 2025.07.13~2025.07.19, 한국해양대학교 전기추진체계실험실

활동 개요: 시뮬레이터 기반 전기추진 시스템의 구성과 내부 통신 흐름을 실습하였다. 추진 제어 장치와 각 구성 모듈 간 데이터 교환 과정을 분석하며 실제 운용 방식에 대해 학습하였다.

성과: 선박의 추진 계통과 네트워크 구조를 이론이 아닌 실습을 통해 심층적으로 파악하였으며, 향후 보안 분석 시 추진 시스템과 네트워크 간 연계 취약점 검토에 활용할 수 있는 기반 지식을 확보하였다.

3-5-9. 한국해양대학교 GMDSS 실습실




**[사진 11,12] 해양대학교 GMDSS 실습 장면**

활동 시기/장소: 2025.07.13~2025.07.19, 한국해양대학교 GMDSS 실습실

활동 개요: 국제표준 GMDSS(Global Maritime Distress and Safety System) 통신 장비(SART, NAVTEX, INMARSAT 등)를 활용한 조작 실습을 진행하였다. 실제 해상 구조·안전 통신 절차와 장비 운용법을 체험하였다.

성과: 실무 통신 환경을 직접 체험하며, 해상 통신 시스템의 취약점과 대응 구조에 대한 인사이트를 확보하였다. 이를 통해 향후 해양 통신 보안 연구에 필요한 기반 지식을 습득하였다.

# 4. 향후 계획

## 4-1. 추후 일정

### 4-1-1. **국제 학술대회 및 컨퍼런스 발표**

- 2025년 12월 영국에서 개최되는 BlackHat Europe 발표 제안서(CFP) 제출 예정
- 오픈소스 프로그램에서 발견한 실제 취약점 사례를 중심으로 발표
- 취약점 발굴 과정, 공격 벡터, 그리고 실효성 있는 대응 방안 제시

### 4-1-2. **해상 사이버보안 분야 최신 위협 동향 및 기술적 대응 전략 공유**

- SCI급 저널 논문 작성 추진
- 기존 논문과 실증 실험 결과를 바탕으로 국제 학술지 투고용 논문 작성·제출

### 4-1-3. **실제 환경 적용 및 검증**

- 실제 프로그램 기반의 다양한 공격 시나리오 실험 환경 구현
- 실제 운항 환경에 근접한 조건에서 보안 기술의 효과성 검증