논문 20-45-03-23 The Journal of Korean Institute of Communications and Information Sciences '20-03 Vol.45 No.03
https://doi.org/10.7840/kics.2020.45.3.628

함정 전투체계 소프트웨어 보안성시험 수행방안 연구

이 철 규•, 김 영 갑°

A Study on Software Security Test of Naval Ship Combat System

Cheol-Gyu Yi•, Young-Gab Kim°

요 약

함정 전투체계는 함정의 다양한 센서 및 무장, 항해지원 장비 등을 컴퓨터 및 네트워크 기반으로 통합한 복합 무기체계로 핵심 기능과 성능이 소프트웨어 중심으로 개발되어 있으며, 이기종 무기체계 간의 통합이 확대됨에 따라 소프트웨어의 복잡도와 연결성이 더욱 증가하고 있다. 그 결과로 소프트웨어 안에 포함된 사이버보안 취약점으로 인한 사이버공격의 위험이 새로운 이슈로 대두되고 있으나, 체계개발 시 사이버보안의 중요성이 체계 자체의 성능과 기능에 비해 상대적으로 낮게 평가되어 있고, 체계적인 소프트웨어 보안성시험이 수행되고 있지 않아 사이버보안에 강건한 체계를 개발하는데 제한되고 있다. 본 논문은 전투체계 소프트웨어에 대한 특성과 현 무기체계 소프트웨어의 개발 관련 규정을 분석하여 사이버보안 강화를 위한 소프트웨어 보안성시험 수행 방안을 제안한다.

키워드 : 무기체계, 함정 전투체계, 개발 보안, 소프트웨어 보안, 보안성 시험
Key Words : Weapon system, Combat system, Development security, Software security, Security test

ABSTRACT

Naval ship combat system is the complex weapon system, which is composed of various sensor, weapon and navigation systems. The core function and performance of combat system is developed by computer software. and expansion of integration of heterogeneous system has been increasing the complexity and connectivity of combat system software. As a result, it raise new issue that software security vulnerabilities will cause the cyber threat but the importance of the cyber security is evaluated lower than that of the function and performance of combat system. Thus, the software security test is not conducted systematically, which restricts to develop the robust cyber security of combat system, This paper proposes the implementation of software security test for naval combat system as it analyze the characteristics of the combat system and regulations of weapon system development.

Ⅰ. 서 론

군의 무기체계는 이미 오래전부터 상용 컴퓨팅 기술을 적용하여 개발해 왔으며, 그 적용 범위와 활용성은 더욱 확대되고 있다. 더욱이 최근 무기체계의 특성은 자동화 및 네트워크를 통한 상호연결성이 높아지

※ 본 연구는 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00498, (창조씨앗2단계) 딥러닝 기반 사용자 행동정보 분석을 통한 인증 및 시스템 내 이상행동 탐지기술 개발)
• First Author : Security Engineering Lab., Dept. of Computer and Information Security, Sejong Univ., naaf0605@empal.com
° Corresponding Author : Security Engineering Lab., Dept. of Computer and Information Security, Sejong Univ., alwaysgabi@sejong.ac.kr, 정회원
논문번호 : 201911-260-C-RE, Received October 29, 2019; Revised December 3, 2019; Accepted December 16, 2019

고 있고[1], 핵심 성능과 기능이 하드웨어에서 소프트웨어 중심으로 변화하면서 소프트웨어의 비중과 중요성이 증가하고 있다[2]. 또한 이기종 무기체계 간의 통합이 확대됨에 따라 소프트웨어의 복잡도와 연결성이 더욱 증가하고 있으며, 그 결과로 무기체계 소프트웨어 안에 포함된 사이버 취약점으로 인한 사이버공격의 위협이 새로운 이슈로 대두되고 있다. 특히, 함정 전투체계는 대표적인 무기체계로 위와 같은 배경으로 인해 다른 무기체계에 비해 더 많은 사이버공격에 노출될 수 있다. 하지만, 함정 전투체계를 포함한 무기체계를 개발 시 사이버보안의 중요성이 무기체계 자체의 성능과 기능에 비해 상대적으로 낮게 평가되어 있으며, 소프트웨어에 대한 체계적인 보안성시험이 수행되지 않아 사이버위협에 강건한 체계를 개발하는데 제한된다.

'18.10월, 미국 회계감사원(GAO)이 보고한 "Weapon System Cyber Security"에 의하면 미군이 개발 중인 무기체계는 사이버공격에 취약하고 기본적인 취약점에 대한 적절한 조치와 사이버보안 기술 적용이 미흡하여 해킹공격의 위험성이 매우 심각하다고 언급하였다[1,3]. 즉, 무기체계 개발자가 체계개발 시 사이버 취약점에 대한 체계적인 검증을 수행하지 않으면, 사이버공격이 가능한 무기체계를 양산할 수밖에 없다.

방위사업청의 "무기체계 소프트웨어 개발 및 관리 매뉴얼"에 의하면, 무기체계에 대한 소프트웨어 보안성시험은 체계개발 단계에서 신뢰성 시험과 함께 수행되는 시험으로 "소프트웨어 구현" 및 "소프트웨어 통합/시험" 절차를 통하여 수행된다. 신뢰성 시험은 소프트웨어의 오류와 결함을 사전에 식별하고, 보안성시험은 행정자치부가 발간한 "소프트웨어 개발보안 가이드"를 적용하여 응용프로그램에 대한 시큐어코딩(Secure Coding) 여부를 확인한다[4].

하지만, 보안성시험 적용 대상이 무기체계 중 전장관리정보체계(C4I체계 등)에만 적용하는 것으로 국한되어 실질적으로 함정이나 항공기 등에 탑재된 무기체계의 소프트웨어 보안성시험은 수행되지 않는 것과 같다.

현재까지 무기체계 소프트웨어에 대한 다양한 신뢰성 시험 연구가 수행되어 시험방법론의 발전과 소프트웨어 품질향상에 많은 기여를 하였으나, 보안성시험은 많은 연구가 수행되지 않았다. 본 논문은 이와 같은 문제점을 개선하고, 체계적인 소프트웨어 보안성시험 수행 방안을 제안하며, 아래와 같이 기여할 것이다.

- 소프트웨어 보안성시험 수행을 위한 구체적인 프레임 정립: 소프트웨어를 구성하는 응용프로그램, 미들웨어 및 운영체제에 대한 시험 수행 분야(Category)와 세부 시험항목을 정립한다.
- 소프트웨어 설계 시 보안 요구조건 반영: 소프트웨어 시험은 소프트웨어 설계 시 중요하게 검토되므로 보안성시험에서 요구하는 보안 요구조건이 설계단계에서 필수적으로 반영하도록 한다.
- 사이버공격에 강건한 전투체계 개발: 체계적인 소프트웨어 보안성시험을 통하여 소프트웨어 안에 내재될 수 있는 취약점을 제거하고 사이버공격에 강한 전투체계를 개발하도록 한다.

본 논문은 Ⅱ장에서 관련연구를 통하여 함정 전투체계 소프트웨어의 특성과 무기체계 소프트웨어의 신뢰성 및 보안성 시험을 살펴보고, 보안성시험의 한계점을 도출하였다. Ⅲ장은 함정 전투체계 특성을 고려한 보안성시험 수행 분야를 정립하고, 세부 시험항목을 도출함으로써 체계적인 보안성시험 수행 방안을 제안하였다. Ⅳ장은 본 논문이 제안한 방안과 기존 전장관리정보체계의 소프트웨어 보안성시험에 적용하는 기준을 비교 분석함으로서 평가하고, Ⅴ장에서 결론을 맺었다.

Ⅱ. 관련연구

2.1 함정 전투체계 소프트웨어

함정 전투체계는 함정에 탑재된 탐지장비, 무장, 항해지원 장비 등을 네트워크로 연결하여 통합된 전술상황 정보를 생산 및 공유하고, 표적의 탐지, 추적, 위협분석, 무장할당, 교전 및 명중평가를 자동화한 무기체계이다[5]. 또한 임무 특성상 다양한 무기체계의 기능과 성능을 융합한 복합체계로 소프트웨어 중심의 통합된 기능과 성능을 발휘한다.

[그림 1]은 전투체계 소프트웨어의 구조를 계층적으로 나타낸 그림으로, 응용프로그램(Application), 미들웨어 및 운영체제로 구성되어 있다. 가장 상위단계에 있는 응용프로그램은 전투임무 수행을 위한 다양한 기능들을 수행하도록 개발된 소프트웨어로 서버(정보처리장치)와 운용자 콘솔 등에서 작동한다. 표적관리, 교전관리, 운용자 GUI환경 등의 전술응용 부분과 공통적으로 운용되는 전술지원 부분으로 탑재되어 기능을 수행하며, 전술응용 프로그램은 모듈화 개념이 적용되어 개발함으로써 부여된 임무와 세부요구 기능에 따라 다양하게 변경될 가능성에 용이하게 대응할 수 있도록 하였다[6,7].

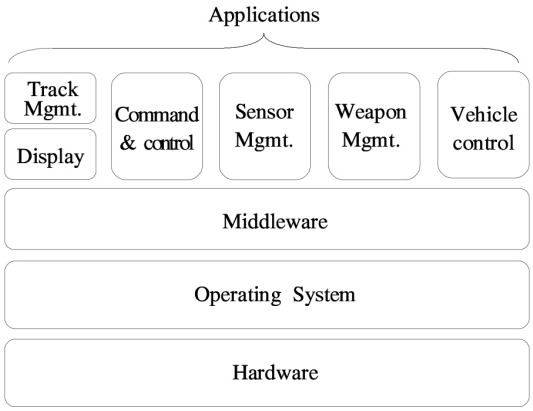

그림 1. 전투체계 소프트웨어 구조[6]
Fig. 1. Combat system software structure[6]

미들웨어는 응용프로그램과 운영체제 중간에 위치하여 공통적인 서비스를 제공하며, OMG(Object Management Group)에서 표준화한 DDS(Data Distribution Service) 미들웨어를 탑재하고 있다. DDS는 통신 미들웨어로 분산된 환경에서 실시간 처리를 위한 발간(publish) - 구독(subscribe) 모델에 기반하여 각 노드 간의 실시간 통신을 지원한다[8,9]. DDS의 보안표준은 2016년 버전 1.0 표준문서를 통하여 발표되었으며, 데이터의 비밀성 및 무결성 보장, 네트워크 참여자의 상호인증 및 상호접근 권한 확인, 부인봉쇄 기능이 제공되었다[10,11].

운영체제는 상용 운영체제를 도입하여 운영하고 있으며, 리눅스, 유닉스, 윈도우 및 실시간 운영체제인 VxWorks 등에 해당된다.

전투체계를 구성하는 각각의 소프트웨어들은 매우 연계성이 높으며 상호의존성이 강하여 운영체제의 설정을 변경 시 응용프로그램의 설계를 변경해야 하는 경우도 발생할 수 있다.

2.2 무기체계 소프트웨어 신뢰성 및 보안성 시험

2.2.1 개념 및 수행절차[4]

무기체계 소프트웨어의 신뢰성 및 보안성시험은 소프트웨어에 대한 오류와 결함을 식별하고, 사이버공격의 원인인 보안약점을 사전에 제거하기 위한 시험으로 기술적인 정확성과 적절성을 가지고 정해진 요구사항을 충족하는지 검증한다. 이 시험은 [그림 2]에서와 같이 무기체계 개발과정 시 체계개발 단계의 "소프트웨어 구현 단계"와 "소프트웨어 통합 및 시험 단계"에서 연구개발주관기관(방산업체 또는 국과연)이 수행하며, 기술지원기관(국과연)은 시험을 지원 및 점검하고 시험결과를 검증하는 역할을 수행한다.

적용대상 소프트웨어는 개발 소프트웨어와 공개 소프트웨어(Open Software) 및 자동생성 코드이며, 시험대상 언어로는 C, C++, JAVA 및 C#을 원칙으로 하고 있다. 소프트웨어 신뢰성 및 보안성시험 항목에 대한 세부계획, 절차 및 결과는 소프트웨어 산출물인 "소프트웨어통합시험계획서(STP)", "소프트웨어통합시험절차서(STD)", "소프트웨어통합시험결과서(STR)"에 포함하여 추적성을 유지한다.

체계개발 단계 이후에 개발시험평가와 운용시험평가를 각각 연구개발주관기관과 소요군이 주관하여 수행하는데, 개발시험평가 단계에서 기술지원기관은 소프트웨어 보안성시험 항목에 대한 내용을 확인한다. 신뢰성 및 보안성시험은 체계개발 단계 내에서 여러 절차와 문서화를 통해 구체화를 진행하고, 체계 설계 및 개발과 유기적인 연계를 통해 이 시험에서 요구하는 사항을 설계 및 개발에 반영하는 것을 알 수 있다.

2.2.2 소프트웨어 신뢰성 시험[4]

소프트웨어 신뢰성 시험은 소프트웨어가 동작할 수 있는 다양한 경우의 수를 확인함으로써 소프트웨어가 일으킬 수 있는 결함을 식별하는 시험으로 정적시험

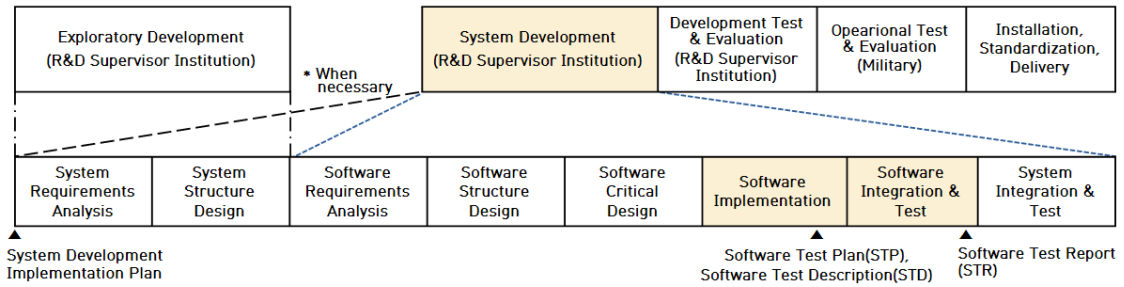

그림 2. 전투체계 소프트웨어 구조무기체계 소프트웨어 개발 절차[4]
Fig. 2. Weapon system software development process[4]

과 동적시험으로 구분하고 있다.

정적시험은 소프트웨어를 실행하지 않은 상태에서 잠재적인 결함을 검출하는 시험으로 코딩규칙(Coding Rule) 검증 및 취약점 점검, 소스코드 매트릭(Code Metric) 검증을 시험한다. 취약점 점검은 미국 국토안보부(국가사이버보안국) 산하 비영리 단체인 MITRE가 제공한 소프트웨어 취약점 표준 목록인 CWE(Common Weakness Enumeration)로 CWE-658, 659, 660을 준수하도록 하고 있으며, C, C++, JAVA 언어로 개발할 때 소스코드에 발생빈도가 높은 취약점 목록들이다.

동적시험은 소프트웨어를 실제 하드웨어에 탑재한 상태에서 소프트웨어 코드 실행율(Code Coverage)을 점검하는 시험이다. 코드 실행율은 동적시험을 결정하는 기준으로 전체 소스코드에서 시험된 코드를 백분율로 나타낸 것이며, 문장 실행율(Statement Coverage), 분기 실행율(Branch Coverage), MC/DC 실행율(Modified Condition/Decision Coverage)로 구분된다. 사업개발담당자는 결함의 발생빈도, 영향성 및 제어가능성을 평가하여 방위사업청의 “무기체계 소프트웨어 개발 및 관리 매뉴얼”에 제시된 등급분류 표에 따라 해당 등급에 맞는 실행율을 선정한다.

2.2.3 소프트웨어 보안성 시험[4,13]

소프트웨어 보안성시험은 행정자치부의 “소프트웨어 개발보안 가이드”를 무기체계 중 전장관리정보체계(C4I체계 등)에만 적용하는 것으로 규정화되어 있으며, 응용프로그램의 시큐어코딩 여부를 확인한다. 이 가이드는 MITRE가 제공하는 보안약점인 CWE와 보안코딩과 관련된 다양한 공식 발표 자료를 참고하여 작성되었으며, 소프트웨어 개발과정에서 개발자의 실수, 논리적인 오류 등으로 발생할 수 있는 보안 취약점을 최소화하여 사이버위협에 대응할 수 있는 안전한 소프트웨어를 개발하기 위한 지침이다.

또한, 소프트웨어의 기밀성, 무결성 및 가용성을 위한 보안통제 기능을 구현하기 위해 체계개발 단계에서 기술적 통제항목에 중점을 두었으며, 설계단계의 보안설계 기준과 구현단계의 보안취약점 제거 기준으로 구분되어 있다.

설계단계의 보안설계 기준은 입력데이터 검증 및 표현, 보안기능, 에러처리, 세션통제 4개 분야의 20개 보안요구 항목으로 정의되어 있으며, 구현단계의 보안취약점 제거 기준은 설계단계의 보안요구 항목과 연계하여 [표 1]과 같이 7개 분야(입력데이터 검증 및 표현, 보안기능, 시간 및 상태, 에러처리, 코드오류, 캡슐화, 세션통제) 47개 항목으로 정의되어 있다.

표 1. 소프트웨어 구현단계 보안취약점 항목
Table 1. Security vulnerability Items of software Implementation phase

Category	Sub-items
Input data verification and presentation	• SQL injection • Cross-site scripting • Path traversal and resource injection • Operating system command injection • Unrestricted upload of files with dangerous form • URL redirection to untrusted site • XQuery injection • Xpath injection • LDAP injection • Cross-site request forgery • HTTP Response Splitting • Uncontrolled format string insertion • Memory buffer overflow • Integer overflow • Reliance on untrusted input a security domain
Security function	• Permission without proper authentication to significant function • Inappropriate authorization • Incorrect permission assignment for critical resource • Use of weak encryption algorithm • Use of inadequate encryption key length • Plaintext storage of significant information • Plaintext transmission of significant information • Use of insufficiently random numbers • Hard coded password • Hard coded encryption key • Use of vulnerable password • Information disclosure due to cookies saved in user HDD • System information in source code comments • Use of a one-way hash without a salt on saving password • Code download of without integrity check • Absence of repetitive authentication attempt limitation
Time and status	• Time of check, time of use race condition • Loop with unreachable exit condition or recursive function
Error processing	• Information disclosure through error message • Detection of error condition without action • Improper processing for unusual or exceptional conditions
Code mistake	• Null pointer dereference • Improper resource shutdown or release • Use After Freed resource

	• Use of Uninitialized Variable
Encapsulation	• Disclosure of data due to wrong session • Leftover debug code • Exposure of system data to an unauthorized control • Private array-typed field returned from a public method • Public data assigned to private array-typed field
API misuse	• Security Decision dependent on DNS Lookup • Use of vulnerable API

하지만, 이 시험분야는 사이버보안 관점의 보안통제 분야보다는 개발자 관점의 분야로 구분되어 있으며, 세부 시험항목들은 전투체계와 관련성이 적은 웹서비스와 DBMS(Database Management System)에 해당하는 내용도 다수 포함하고 있다.

소프트웨어 신뢰성시험과 보안성시험은 체계개발 단계에서 각 시험이 요구하는 항목을 체계개발에 반영하도록 하여 소프트웨어의 품질과 유지보수성을 향상하고 사이버보안을 강요하도록 한다. [표 2]는 신뢰성시험과 보안성시험에 대한 목적과 기능, 적용대상, 평가기준 등을 비교한 내용이다.

표 2. 소프트웨어 신뢰성 및 보안성 시험 비교
Table 2. Comparison between reliability and security test of software

	Reliability test	Security test
Purpose	• Quality, maintenance, and efficiency of software	• Response to cyber threat
Function	• Identify software fault	• Preliminary delete of security weakness
Applicable target	• Weapon systems	• C4I systems
Evaluation criteria	• Static test : check if there is fault detection • Dynamic test : check if it achieves 100% code coverage	• Check if observed "Guideline for Software Development Security"
Contents	• Coding rule and vulnerability check, Matrix verification • Code coverage	• Check security requirements of application on software design and implementation phase

2.3 소프트웨어 보안성시험의 한계

소프트웨어 보안성시험은 신뢰성시험과 더불어 체계개발 단계에서 개발주관기관의 주관으로 수행되는 시험으로 행정자치부의 "소프트웨어 개발보안 가이드[13]"를 적용하는 것으로 한정하고 있다. 또한 이 가이드가 제시하는 실제 시험 항목은 응용프로그램에만 해당되고 그와 연관성이 높은 미들웨어와 운영체제에는 적용되지 않는 것을 확인할 수 있다.

[그림 1]에서와 같이 전투체계 소프트웨어 구조상 최상단에 있는 응용프로그램의 취약점이 제거되었다고 하더라도 운영체제와 미들웨어의 취약점이 보완되지 않은 상태로 남아 있다면 사이버공격에 안전할 수 없다. 즉, 무기체계에 대한 보안성시험은 응용프로그램뿐만 아니라 운영체제 및 미들웨어에 대해서도 체계적으로 수행되어야 한다.

또한, 시험항목 중 많은 부분이 웹서비스와 DBMS를 대상으로 하고 있다. 하지만, 함정 전투체계는 웹서비스와 DBMS 기능을 제공하지 않으며, 분산구조의 클라이언트/서버 환경으로 되어 있으므로, 이 가이드가 제시하는 보안성시험 항목을 전투체계 소프트웨어에 그대로 적용할 수 없다.

Ⅲ. 전투체계 소프트웨어 보안성시험 향상 방안

함정 전투체계 소프트웨어의 보안성시험은 응용프로그램뿐만 아니라, 그와 밀접한 연관이 있는 미들웨어 및 운영체제에도 적용하여 체계적인 보안성 검증을 수행해야 한다. 응용프로그램은 미들웨어 및 운영체제의 특성과 보안 설정에 따라 개발 방향이 변경될 수 있으므로 미들웨어와 운영체제의 보안성시험은 응용프로그램과 함께 통합적으로 수행될 수 있도록 함으로써, 소프트웨어 설계 단계부터 각 소프트웨어 간에 사이버보안의 연계성을 유지하도록 해야 한다.

보안성시험 향상을 위한 기본방향은 아래와 같다.

- 사이버보안 관점에서 보안성시험 수행 분야를 정립한다. "소프트웨어 개발보안 가이드[13]"가 제시한 보안성시험 분야는 개발자 관점에서 시험 분야가 구분되어 있으므로, 보안통제 사항을 체계적으로 반영하여 시험하기에는 제한이 있다. 그러므로 사이버보안 관점에서 보안성시험 분야를 정립함으로써 체계적인 시험이 수행될 수 있는 프레임을 제공한다.
- 응용프로그램에 대한 보안성시험 세부 항목을 최적화 한다. 전투체계의 특성과 환경을 고려하여 "소프트웨어 개발보안 가이드[13]"가 정의한 구현

단계의 보안취약점 제거 기준을 전투체계의 특성에 맞도록 최적화하고 추가적으로 필요한 보안통제 항목을 반영한다.

- 운영체제와 미들웨어에 해당하는 보안성시험 세부 항목을 도출하여 적용한다. 각 세부 항목들은 전투체계의 특성과 환경을 고려하여 앞에서 정립한 보안성시험 분야를 기반으로 도출한다. 운영체제의 보안성시험 세부 항목은 과학기술정보통신부와 한국인터넷진흥원이 발간한 "주요정보통신기반시설 기술적 취약점 분석/평가방법 상세가이드[14]"를 참고하였다. 이 가이드는 시스템에 대한 기술적인 보안설정 점검을 통하여 취약 여부를 평가하는 지침으로 313개의 점검항목이 정의되어 있다. 미들웨어는 OMG에서 제시한 DDS 보안표준을 참고하였다.

3.1 보안성시험 수행 분야 정립

보안성시험 수행 분야는 기존의 개발자 관점이 아닌 사이버보안 측면의 통제 범위를 고려하여 [그림 3]과 같이 10개의 시험분야로 정립하고, 전투체계 소프트웨어를 구성하는 응용프로그램, 미들웨어 및 운영체제에 해당하는 분야를 적용하였다.

이 시험 수행 분야는 전투체계 소프트웨어가 기본적으로 갖추어야할 사이버보안 통제 분야로 개발자 관점이 아닌 사이버보안 관점에서 시험분야를 통합 및 분류하고, 반드시 필요한 사이버보안 통제 분야를 추가함으로써 그에 따른 세부 시험항목이 체계적으로 선별될 수 있도록 하였다. 이 보안성시험 수행 분야는 응용 프로그램, 미들웨어 및 운영체제에 해당하는 보안성시험을 체계적으로 수행할 수 있는 지표가 된다.

3.2 소프트웨어별 보안성 시험 세부항목 도출

응용프로그램, 미들웨어 및 운영체제 각각의 특성과 기능에 따라, 보안성시험 분야가 요구하는 세부 시험항목을 도출하였다. [그림 3]에서와 같이 각각의 소프트웨어별 시험수행 분야는 같을 수 있지만, 해당 소프트웨어의 특성에 따라 세부 시험항목들이 상이하게 존재한다.

3.2.1 응용프로그램

응용프로그램은 "입력데이터 통제" 등 9개 분야 37개 세부 시험항목으로 구성하였다. 세부 시험항목들은 전투체계 특성과 환경을 고려하여, "소프트웨어 개발 보안 가이드[13]"에서 제시된 항목 중 전투체계에 부합되지 않는 웹서비스와 DBMS 등의 시험항목을 제외

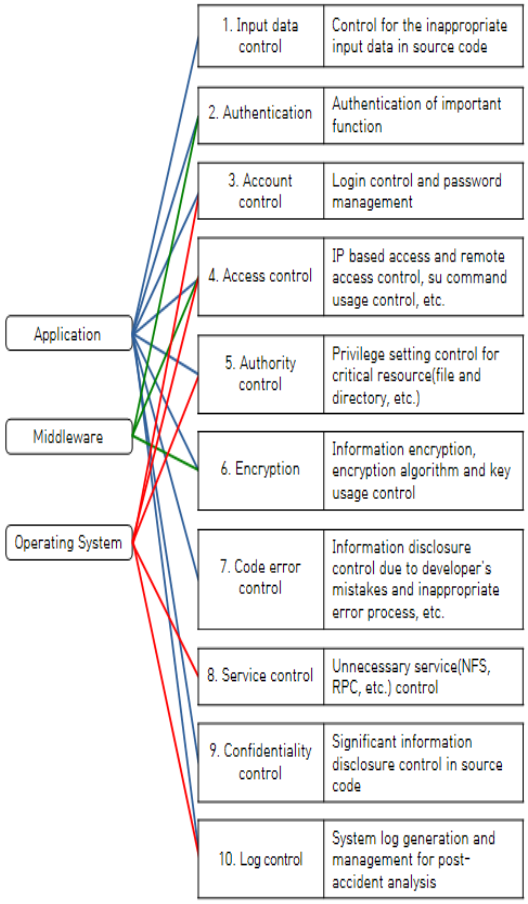

그림 3. 전투체계 소프트웨어 보안성시험 수행 분야
Fig. 3. Combat system software security test categories

하고 최적화하였으며, 전투체계의 중요 전술기능에 대한 접근통제와 중요 시스템 로깅 수행여부를 확인할 수 있는 로그통제 분야를 추가하였다. 이 세부 시험항목들은 소프트웨어 시험 시 소스코드 안에 알려진 취약점이 포함되었는지를 검증한다.

[표 3]은 응용프로그램에 대한 보안성시험 세부 항목으로, "1. 입력데이터 통제" 분야는 부적절한 입력과 버퍼오버플로우를 차단하기 위한 시험이며, "2. 인증", "3. 계정통제", "4. 접근통제", "5. 권한통제"는 부적절한 접근과 권한설정 여부를 검증한다. "6. 암호와"와 "9. 비밀성 통제"는 취약한 암호와 키의 사용 및 중요 정보의 평문 전송이나 소스코드 상에 노출 여부를 확인하고, "7. 코드오류 통제"는 개발자의 실수로 인해 잘못된 함수의 사용 및 의도되지 않은 정보노출 등이 존재하는지를 검증한다. "10. 로그통제"는 중요 보안 데이터(로그인, 접근, 전술정보 처리 등)에 대한 감사로그 생성 및 관리여부를 확인한다.

표 3. 응용프로그램 보안성시험 세부 항목
Table 3. Sub-items of Application security test

Category	Sub-items
1. Input data control (6 items)	① Path traversal and resource injection ② Operating system command injection ③ Integer overflow ④ Uncontrolled format string insertion ⑤ Memory buffer overflow ⑥ Reliance on untrusted input a security decision
2. Authentication (2 items)	⑦ Permission without proper authentication to significant function ⑧ Inappropriate authentication
3. Account control (2 items)	⑨ Use of vulnerable password ⑩ Use of one-way hash without a salt on saving password
4. Access control (1 item)	⑪ Access control to significant tactical function(weapon, track management and control, etc.)
5. Authority control (1 item)	⑫ Incorrect permission assignment for critical resource
6. Encryption (5 items)	⑬ Use of weak encryption algorithm ⑭ Use of inadequate encryption key length ⑮ Plaintext storage of significant information ⑯ Plaintext transmission of significant information ⑰ Use of insufficient random numbers
7. Code error control (16 items)	⑱ Code download without integrity check ⑲ Time of check, time of use race condition ⑳ Loop with unreachable exit condition or recursive function ㉑ Information disclosure through error message ㉒ Detection of error condition without action ㉓ Improper processing for unusual or exceptional conditions ㉔ Null Pointer dereference ㉕ Improper resource shutdown or release ㉖ Use after freed release ㉗ Use of uninitialized variable ㉘ Disclosure of data due to wrong session ㉙ Leftover debug code ㉚ Exposure of system data to an unauthorized control ㉛ Private array-typed field returned from a public method ㉜ Public data assigned to private array-typed field ㉝ Use of vulnerable API
9. Confidentiality control (3 items)	㉞ Hard coded password in source code ㉟ Hard coded encryption key in source code ㊱ Significant system information(ID, password, etc.) disclosure through comments
10. Log control (1 item)	㊲ Generation and storage of audit log related to significant security data (log-in, access, tactical information processing, etc.)

3.2.2 미들웨어

미들웨어는 "인증", "접근통제", "암호화" 3개 분야 3개 세부 시험항목으로 구성하였다. 이 시험항목들은 전투체계 미들웨어의 특성인 실시간성과 성능저하 요인을 고려하여 미들웨어에 반드시 필요한 보안성 시험을 수행하도록 하였다.

[표 4]는 미들웨어 보안성시험 세부 항목으로 도메인 내에서 적절한 인증을 받은 참여자가 정보에 접근할 수 있는지, 중요 데이터에 대한 암호화가 수행되었는지를 검증한다.

표 4. 미들웨어 보안성시험 세부 항목
Table 4. Sub-items of middleware security test

Category	Sub-items
1. Authentication (1 item)	① Appropriate authentication for domain participants
2. Access control(1 item)	② Access control for domain participants
3. Encryption (1 item)	③ Encryption for the transmitted message

3.2.3 운영체제

운영체제는 "계정통제" 등 5개 분야 30개 세부 시험항목으로 정립하였으며, 운영체제에 해당하는 보안설정이 체계적으로 반영되었는지를 검증한다. 특히, 운영체제는 알려진 취약점이 적절하게 조치되지 않을 경우 공격자가 쉽게 접근하여 권한을 탈취할 수 있으므로 보안설정의 검증은 매우 중요하다. [표 5]는 운영체제에 대한 보안성시험 세부항목을 나타내고 있다.

"3. 계정통제"는 계정과 비밀번호에 대한 부적절한 사용 여부를 검증하고, "4. 접근통제"와 "5. 권한통제"는 전투체계에 대한 악의적인 원격접근을 차단 여부와 운영체제 내 중요 디렉터리 및 파일에 대한 부적절한 권한 설정으로 악용될 수 있는지를 검증한다. "8. 서비스 통제"는 해킹공격에 활용될 수 있는 불필요 서

표 5. 운영체제 보안성시험 세부 항목
Table 5. Sub-items of operating system security test

Category	Sub-items
3. Account control (5 items)	① Unnecessary and default account delete ② Vulnerable password check (use numbers, characters, special characters mix) ③ Log-in security check[windows] ④ Administrator and guest account control [windows] ⑤ Anonymous account control in administrator group[windows]
4. Access control (5 items)	⑥ Control of “root” account remote access ⑦ Control of “su” command use by regular users ⑧ Control of accessible IP and port ⑨ Security management of remote access [windows] ⑩ Remote access control to SAM(Security account management) file[windows]
5. Authority control (8 items)	⑪ Setting UMASK, PATH to files ⑫ Setting access privilege to user home and significant directories ⑬ Check if normal files are deleted in dev/directory ⑭ Check if files and directories without owner are deleted ⑮ Setting owner and privilege to significant system files ⑯ Check if there exist files with unnecessary SUID, GUID setting ⑰ Do not allow anonymous enumeration of SAM accounts and shares[windows] ⑱ Directory security management
8. Service control (11 items)	⑲ Check unnecessary services (SNMP, ftp, etc.) ⑳ Setting owner and privilege to “cron” files ㉑ NFS/RPC services stop or security setting ㉒ Setting stop to vulnerable deamon(finger, r-commands, exec, etc.) in inetd.d file ㉓ Stop unnecessary schedule jobs ㉔ Delete of shared folder[windows] ㉕ Setting registry to defend DoS attack [windows] ㉖ Setting console firewall[windows] ㉗ Restrict remote support function[windows] ㉘ Check if console security programs (USB control, NAC, etc.) are installed[windows] ㉙ Delete of unnecessary applications (MS-Office, word processor, etc.) [windows]
10. Log control (1 item)	㉚ Setting significant system logs storage

비스와 프로그램이 작동중인지를 확인하고, “10. 로그 통제”는 해킹공격에 대한 추적 및 사후분석을 할 수 있도록 로그 저장기능이 설정되었는지를 검증한다. 세부 시험항목들 중 윈도우즈 운영체제에 해당하는 항목들은 [Windows]로 표기하였다.

Ⅳ. 제안한 보안성시험 향상 방안 평가

지금까지 함정 전투체계 소프트웨어에 대한 향상된 보안성시험 수행 방안을 제안하였다. 이 방안은 지금까지 소프트웨어 개발단계에서 수행하지 않았던 소프트웨어 보안성시험을 체계적으로 수행할 수 있는 프레임워크를 제공한다.

제안한 보안성시험 수행 방안에 대한 분석은 기존에 무기체계 중 전장관리정보체계(C4I체계 등) 개발 시 보안성시험 수행에 기준이 되는 “소프트웨어 개발 보안 가이드[13]”를 대상으로 비교하였다. 실제 함정 전투체계 개발 시 이 가이드를 적용하는 것은 강제성은 없지만, 본 논문에서 제안한 시험 수행 방안을 비교 분석하기 위한 대상으로 정하였다.

[표 6]은 제안한 보안성시험 비교 분석에서 보는 바와 같이, 우선 보안성시험 적용대상은 기존 응용프로그램뿐만 아니라 미들웨어 및 운영체제에도 적용하여 전투체계를 구성하는 소프트웨어 전반에 대한 보안성시험을 수행할 수 있다. 적용기준은 전투체계의 특성과 환경을 고려하여, 전투체계에 최적화된 시험수행 분야와 분야별로 해당하는 세부 시험항목을 기준으로

표 6. 제안한 보안성시험 비교 분석
Table 6. Analysis between existing test and proposed test

	Existing security test	Proposed security test
Target software	Application	Application, middleware, operating system
Applicable criteria	None (Apply “Software development guideline” to C4I system only)	10 test categories & 70 sub-items
Test categories and sub-items validity	Insufficient (Software developer-centric categories)	Sufficient (Cyber security-centric categories)
Response to cyber threat	Application only	Application, middleware, operating system

적용함으로써 체계적인 시험을 수행할 수 있다.

시험분야 및 세부 시험항목의 타당성은 개발자 관점이 아닌 사이버보안 관점의 시험분야로 정립하고, 응용프로그램, 미들웨어 및 운영체제에게 체계적으로 적용할 수 있는 타당한 세부시험 항목을 도출함으로써 향후 새로운 취약점이 발견되면 정립된 시험분야를 기준으로 타당한 세부 시험항목을 체계적이고 지속적으로 개발하여 보완할 수 있다.

마지막으로 사이버위협 대응은 기존에는 응용프로그램에 대해서만 가능하였으나, 제안한 보안성시험 수행 방안은 전투체계를 구성하는 소프트웨어(응용프로그램, 미들웨어, 운영체제)에 대하여 모두 대응할 수 있다.

V. 결 론

함정 전투체계는 다양한 무장과 센서, 항해지원 장비들을 네트워크와 소프트웨어 기반으로 통합한 복합무기체계로 주요 기능과 성능이 소프트웨어 중심으로 개발되었으며, 최근 들어 상용 소프트웨어 적용의 비중도 증가하고 있다. 하지만, 소프트웨어에 대한 체계적인 보안성시험이 수행되지 않아 사이버위협에 강건한 전투체계를 개발하는데 한계가 있다.

이러한 문제를 해결하기 위해 본 논문은 전투체계의 특성과 환경을 고려하여, 사이버위협에 효과적으로 대응할 수 있는 전투체계 개발을 위한 소프트웨어 보안성시험 수행방안을 제안하였다.

우선, 전투체계 소프트웨어를 구성하는 응용프로그램, 미들웨어, 운영체제에 대한 보안성시험 수행분야를 사이버보안의 관점에서 10개 분야로 정립하여 체계적인 보안성시험을 수행할 수 있는 지표를 제시하였다. 이 시험수행 분야는 전투체계 소프트웨어가 갖추어야할 사이버보안 통제 분야로 이 분야에 따라 세부 시험항목이 체계적으로 선별되고 적용될 수 있도록 하였다.

그리고 각 전투체계 소프트웨어에 해당하는 분야별 보안성시험 세부 항목들을 도출 및 최적화하였다. 이 세부시험 항목들은 전투성능을 최우선으로 하는 전투체계의 임무 특성과 각 소프트웨어가 제공하는 서비스의 특성을 고려하여, 반드시 시험이 필요한 내용으로 최적화함으로써 전투체계 소프트웨어 설계부터 시험에서 요구하는 사항을 반영하도록 하였다.

앞으로 함정 전투체계와 같은 무기체계를 대상으로 하는 사이버위협이 지속적으로 나타날 것이다. 이러한 사이버위협 요소를 분석하고, 그에 타당한 대응 시스템을 연구개발하여 무기체계에 구축해야한다. 단, 무기체계의 임무 특성상 전투성능과 운용성에 영향을 최소화할 수 있도록 구축해야할 것이다. 또한 무기체계 소프트웨어에 대한 보안성시험을 수행할 수 있는 지침이 규정화되어 사이버보안이 강화된 무기체계가 개발되어야 한다.

References

[1] Government Accountability Office, *Weapon Systems Cyber Security,* 2018.

[2] K.-Y. Heo, K.-Y. Kwon, and T.-S. Kim, "A study on the promotion of reliability test for imbedded software of weapon system," *J. Korean Soc. Systems Eng.,* vol. 11, no. 1, pp. 66, Jun. 2015.

[3] J.-K. Lee, "Trend of weapon system cyber security policy," *J. Korea Inst. Inf. Secur. and Cryptology*, vol. 28, no. 6, p. 84, Dec. 2018.

[4] DAPA, *Manual for Development and Management of Weapon System Software*, 2017.

[5] DTaQ, *Dictionary of Military Technical Terms*, 2018.

[6] ROK Naval Education and Training Command, *Combat System Principle and Understanding*, 2016.

[7] K. Emery, *Surface Navy Combat System Engineering Strategy*, US Navy PEO IWG, 2010.

[8] J.-H. Han, "Message encryption methods for DDS security performance improvement," *J. Korea Inst. Inf. and Commun. Eng.*, vol. 22. no. 11, p. 1555, Nov. 2018.

[9] *What is DDS?*, from https://www.dds-foundation.org/

[10] Y.-K. Go and C.-S. Kim, "Cryptographic overhead of DDS security for naval combat system security," in *Proc. KIISE Korea Computer Congress 2017*, p. 1217, Jun. 2017.

[11] *DDS Security Standard Document*, from http://www.omg.org/spec/DDS- SECURITY/1.0

[12] T.-H. Lee, O.-H. Paek, and T.-H. Kim, "Status and improvement direction about weapon system software reliability test," *J. Korea Inst. Inf. Secur. and Cryptology*, vol. 28, no. 6, pp.

77-80, Dec. 2018.
[13] Ministry of the Interior, KISA, *Guideline for Software Development Security*, 2017.
[14] KISA, *Guideline for Analysis and Assess on Technical Vulnerability of Main ICT Infrastructures*, 2017.

이 철 규 (Cheol-Gyu Yi)

1995년 3월 : 해군사관학교 조선공학과 학사
2002년 3월 : 연세대학교 컴퓨터과학과 석사
2019년 3월~현재 : 세종대학교 정보보호학과 박사과정
<관심분야> 무기체계 보안, 사이버보안 관제, 위협헌팅, 전투체계
[ORCID:0000-0003-3006-5078]

김 영 갑 (Young-Gab Kim)

2001년 8월 : 고려대학교 컴퓨터학과부전공
2003년 8월 : 고려대학교 컴퓨터학과 석사
2006년 8월 : 고려대학교 컴퓨터학과 박사
2006년 8월~2008년 3월 : 고려대학교 정보보호대학원 연구교수
2008년 3월~2010년 1월 : 국가평생교육진흥원 선임전문원
2010년 2월~2015년 2월 : 고려대학교 연구교수
2013년 3월~2015년 2월 : 대구카톨릭대학교 IT공학부 조교수
2015년 3월~현재 : 세종대학교 정보보호학과 부교수
<관심분야> 사물인터넷 보안, 보안공학, 위험분석
[ORCID:0000-0001-9585-8808]