미 국방 소프트웨어 개발의 개방형 기술 개발(OTD) 동향

정유진 류원옥* 정해원* 소서영** 강신각*
카이스트 전산학부 박사과정
한국전자통신연구원 책임연구원 *
대전보건대학교 교수 **

I. 서론

세계적으로 오픈소스 소프트웨어(Open Source Software: OSS)에 대한 관심이 커지면서 우리나라를 포함한 각국 정부에서는 OSS 활용 방안을 연구하고 있다. OSS의 장점은 비용 절감, 오픈소스 SW의 품질, 효율적인 개발 및 생산성 향상이다.

오픈소스는 복제, 설치, 운영, 수정, 배포의 자유가 있으나 특정 OSS 라이선스의 소스코드 공개 의무가 있고, 공개되어 있는 소스코드에는 저작권법에 따른 법적 권리 보호가 있고, 오픈소스의 내부사용은 무료이나 외부 배포에 따른 라이선스 의무사항 준수 의무를 갖는 양면성이 있다[1].

현재 OSS 사용이 증가하면서 오픈소스의 보안취약점도 증가하고 있는 추세이나[2] 많은 대부분의 기업의 경우에 오픈소스에 대한 보안 관리 전담팀 또는 보안 전문가가 없으며, 보안 경고 및 알림 체계도 없고, 보안 패치 업데이트도 거의 수행하지 않는 것이 현실이다[1]. 그러나 안전한 오픈소스 사용을 위해서는 오픈소스 버전 사용 정책 및 지속적인 모니터링으로 안전한 패치 버전을 제공해야 오픈소스 사용에 따른 위험을 최소화할 수 있다.

OSS가 ICT 산업에 미치는 영향력이 증가하면서 세계적으로도 관심이 커지고 있다. 한국도 역시 정부·공공부문을 비롯하여 금융과 제조 등 각 산업부문에서 OSS 도입을 적극 검토중이다[7]. 국내 OSS 시장은 2015년부터 2020년까지 연평균 15% 이상 성장하여 2020년에는 시장 규모가 2,800억 원을 초과할 것으로 예상되고 있다[1]. 이에 따라 국방부는 2015년 2월에 미래창조과학부(현 과학기술정보통신부)와 OSS, 사물인터넷 관련 기술 개발 및 활용 촉진을 위한 양해각서를 체결하였으며, 이를 통해 국방 ICT 분야에 OSS를 적극적으로 도입·활용하여 외국 SW기업에 대한

* 본 내용은 정유진 수석연구원(042-860-4886, parvaavis867@gmail.com)에게 문의하시기 바랍니다.
** 본 내용은 필자의 주관적인 의견이며 IITP의 공식적인 입장이 아님을 밝힙니다.

의존성을 낮추고, 군 전문인력들의 SW 개발·유지관리 역량을 강화시켜 나가기로 하였다[7].

OSS 사용 선도국인 미국에서는 OSS의 중요성을 인식하고 국방 분야에서도 OSS를 활용하는 다양한 정책을 수행하고 있다. 특히, 보안성을 높이기 위해 DSS(Defense Security Service)를 중심으로 다양한 프로젝트를 도입하여 전 세계 개발자로부터 문제점을 수집하여 취약점을 개선하고 있다. 미 국방성(Department of Defense: DoD)에서는 국방 소프트웨어 개발 시에 OSS의 개방형 개발 방식을 도입한 개방형 기술 개발(Open Technology Development: OTD) 방식을 적용하고 있다.

"Open Technology Development: Lessons Learned and Best Practices(2011)[6]" 문서를 통해 국방 소프트웨어 개발을 위한 개방형 기술 개발법(OTD)에 대해 소개를 하고 있다. 이 문서를 통해 미국의 OSS에 대한 정책과 개방형 기술 개발, 그리고 이에 관련된 법적 관계 사항을 확인할 수 있다. 미국의 개방형 기술 개발법 사례를 통해 군 소프트웨어의 공개 개발 절차가 어떻게 진행되는지 알아보기 전에 우리나라 군 소프트웨어에서의 OSS 활용 및 국방부에서 추진한 OSS 관련 사례를 살펴보면 다음과 같다.

1. 우리나라 국방 관련 오픈소스SW 가이드라인

국방부에서는 "무기체계 소프트웨어 개발 및 관리 매뉴얼"[3]을 통해 OSS에 대한 적용 가이드라인을 제시하고 있다. OSS 지적재산권과 관련하여 "OSS가 포함된 경우에는 지식재산권 및 라이선스에 저촉되지 않도록 관리한다", 또한 "연구개발 시 소스코드 공개 의무가 있는 공개 소프트웨어(Open Source)를 사용해서는 안 된다"는 규정을 정의하고 있다[3].

그리고 무기체계 연구개발 시 라이선스 적용기준은 다음과 같이 규정하고 있다.

① 소스코드 공개 의무가 있는 OSS를 사용해서는 안 된다.
② 연구개발 주관기관은 OSS 사용 시 특허권 등 권리관계의 포함 여부를 사전에 확인하여 지식재산권 분쟁이 발생되지 않도록 관리해야 한다.
③ SW에 관한 지식재산권과 OSS 라이선스 의무사항을 준수한다[4].

보안성 특징과 관련해서는 "전장관리정보체계에 사용되는 OSS에 대해서는 보안성 시험을 수행하는 것을 원칙으로 한다. 적용된 OSS에 대한 보안취약점 패치 등을 형상관리 절차를 통해 시행한다."고 언급되어 있다[4].

그리고 과제 추진과 관련하여 연구개발 주기에 따른 OSS 식별 및 적용 계획 작성, OSS 사용 계획 작성 및 검토, 공개SW에 대한 라이선스 준수여부를 확인, 사용한 공개SW에 대한 위험분석, 프로그램

소스코드와 라이브러리에 대한 라이선스 준수여부 확인, 공개SW에 대한 신뢰성 시험 및 보안성 시험, 공개의무가 있는 소프트웨어 사용 여부 확인 등을 수행해야 한다고 상세하게 언급하고 있다[4].

이러한 규정을 기반으로 OSS를 관리하고 있으며, OSS에 대한 정부의 도입 시도에 맞춰 국방부도 여러 분야에서 OSS 도입 방안을 적극적으로 검토하고 있다[7].

국방부에서 추진한 개방형 OS 관련 사례를 살펴보면 개방형 운영체제(OS), 하모니카 OS, 구름 플랫폼, 국방부 오픈소스 아카데미 등이 있다. 공개SW 기술 개발로 핵심SW를 국산화하고, 보안 프레임워크 기술 개발과 내재화도 이루어질 것으로 전망된다[5].

2. 개방형 OS

국방부에서 개방형 OS를 적용하기 위한 사업을 추진 중에 있다. 소프트웨어정책연구소에 따르면[8] 현재 군에서 사용 중인 PC의 윈도 제품 의존도가 99.99%에 이른다. 높은 OS 종속성으로 인한 SW 종속성 심화와 과다한 예산 소요 문제를 줄이고자 국방부에서는 현 OS에서 개방형 OS로의 전환을 시도하는 것으로 해석된다. 우선적으로 병사들의 병영생활 개선을 위해 운영중인 사이버 지식정보방 PC 교체 물량의 50% 정도를 개방형 OS로 도입하는 사업을 추진하고 있다.

3. 하모니카 OS

하모니카(HamoniKR)는 2014년 초 발표된 미래창조과학부의 공개SW활성화계획의 일환으로 미래창조과학부와 정보통신산업진흥원의 지원사업으로 개발된 리눅스 기반 개방형 운영체제이다. 2016년 이후로는 하모니카 커뮤니티(hamonikr.org) 참여자들 중심으로 자발적으로 수정, 배포 활동이 이루어지고 있다. 하모니카는 한글화 서비스, 사용자 인터페이스, 사용자경험 등을 중점적으로 개선하여 별도의 한글 설정 없이 바로 설치하여 사용할 수 있다는 장점이 존재한다. 그러나 하모니카에 대한 인지도 부족 및 다른 응용 SW와의 호환성 문제 등으로 보급이 한정되었다

국방부에서는 2016년 개방형 OS 도입 효과를 평가하고자 육군, 해군, 공군 8개 부대에 하모니카 OS를 설치하고 적용성을 검토한 적이 있다. 시행 부대를 대상으로 한 설문 조사 결과, 개방형 OS의 사용성이 윈도 OS 사용성보다 비슷하거나 더 높은 것으로 나타났다. 특이하였던 점은 윈도 대비 하모니카의 사용자 인터페이스(User Interface: UI)가 편리하고 접근성이나 속도 면에서 우수하다는 평가를 받았다.

4. 구름 플랫폼

과학기술정보통신부와 한국전자통신연구원 산하 국가보안기술연구소가 2015년부터 윈도 독점 상황 개선과 하모니카의 문제점을 개선하고 보안 기능이 강화된 개방형 OS인 구름 플랫폼을 개발하였다.

2018년 5월, 해군사관학교는 구름 플랫폼을 활용한 클라우드 기반 원격교육시스템 구축 사업을 수행하였다. 해당 사업을 통해 클라우드 기반 원격교육 시스템을 구축하여 원격 강의 및 콘텐츠를 활용한 교육으로 교수인력 부족현상을 해결하고자 하였다. 또한, 공급자(Vendor) 독립적인 개방형 OS 사용으로 클라우드 기반의 안전한 군 업무환경 도입 가능성을 확인했다.

5. 국방부 오픈소스 아카데미

국방부와 과학기술정보통신부가 군 오픈소스 역량 강화를 위해 국방부 오픈소스 아카데미를 개설하였다. 오픈소스를 배우고자 하는 대한민국 장병 누구나 수강할 수 있다. OSS 공통과정을 비롯한 온라인 콘텐츠와 오프라인 콘텐츠 모두 제공한다. 온라인 콘텐츠의 경우 기술 분야별 코스로 구성되어 있으며, 오프라인 콘텐츠는 기술세미나와 실습기반의 소프트웨어 개발 캠프로 구성되어 있다.

II. 개방형 기술 개발 소개

1. 개방형 기술 개발(OTD)

소프트웨어가 군에서 주요한 위치를 차지하게 됨에 따라서, 주어진 미션에 빠르게 적응할 수 있는 소프트웨어 개발의 필요성이 대두되었다. 기존의 개발 방법(정부 단독 개발/특정 기업의 독점적 개발)은 소프트웨어 최종 목표에는 최적화되어 있지만, 전체 측면에서 살펴볼 때 경우에 따라서는 비효율을 초래할 수도 있다. 특히, 유지보수 과정이 독점적으로 이루어지므로 경쟁사가 더 나은 업데이트를 제시할 수 없다. 이는 유지보수 과정에서 경쟁에 의한 품질 개선 효과를 볼 수 없게 됨을 뜻한다. 결국은 경쟁력 상실과 특정 벤더에 대한 의존성으로 인해 유지보수 시 추가적인 비용이 발생한다. 이러한 문제를 해결하기 위해 DoD에서는 기존과는 다른 방식의 개발 전략을 필요로 하게 되었다.

개방형 기술 개발(OTD)은 여러 단체의 개발자(정부 및 군 외부의 단체)가 소프트웨어나 시스템을

분산적으로 개발하고 유지하는 개발 접근법이다. 2006년 DoD에서 발표한 OTD 로드맵 계획(APR 2006)[9]에 따르면 OTD는 다음과 같은 네 가지 분야에서의 두드러진 발전 사항을 결합한다.

- 개방형 표준(Open Standards)과 인터페이스
- OSS와 설계
- 협업/분업 문화와 온라인 지원 툴
- 애자일(Agility)[1] 기반 기술

[표 1]에서 각 네 가지 분야에서의 사용/확장/생성에 관한 사항을 확인할 수 있다.

[표 1] OTD 핵심 요소

OTD 핵심 요소	사용(Use)	확장(Extend)	생성(Create)
개방형 표준과 인터페이스	전용 표준보다 개방형 표준/인터페이스/데이터 형식 사용을 선호, 구현된 대안 품목의 사용 가능성 시험을 위한 테스트 사용 가능	표준이 요구사항을 충족하지 않는다면 해당 개방형 표준 관리자에게 변경 제안사항 제출	해당되는 경우 새로운 개방형 표준을 시작
OSS와 설계	확장 OSS의 사용	코드 베이스를 포크(fork)주) 하지 않고 기존 OSS에 수정	새 OSS 프로젝트 시작
협업/분업 문화와 온라인 지원 툴	기존 커뮤니티의 공동 프로젝트 관리 툴 사용에 참여	기존 협업 프로젝트 관리 툴 혹은 인프라(기밀 소프트웨어 개발)를 확장	새로운 조직 간 협업 프로젝트 관리 툴/인프라 구축. 개방적 정부 기성품(OGOTS) 혹은 OSS 가능
애자일 기반 기술 (개방형 시스템/개방형 아키텍처 포함)	시스템의 모듈화 정도 혹은 대응성이 뛰어난 제품 및 서비스 사용 권장, 단일 플랫폼이거나 폐쇄형인 제품 및 서비스 사용을 지양	사용 중인 제품과 서비스의 모듈화, 플랫폼 지원 및 맞춤 확장 가능성 향상	변화 대응성이 높고 시스템의 모듈화 정도가 높은 제품, 서비스 혹은 인프라를 제작

주) 포크(fork) 또는 소프트웨어 개발 포크, 프로젝트 포크. 소프트웨어 소스 코드를 통째로 복사하여 독립적인 새로운 소프트웨어를 개발하는 것

〈자료〉 Scott, Wheeler, Lucas, &Herz, "Open Technology Development(OTD): Lessons Learned and Best Practices for Military Software," 2011, p.35.

OTD는 OSS와 개방형 정부 기성품(Open Government Off-the-shelf: OGOTS) 개발을 모두 포함하는 개념이다.

DoD에서는 OSS를 "사용, 연구, 재사용, 수정, 개량 및 재분배에 사용할 수 있는 인간이 판독 가능한 소스 코드를 사용하는 소프트웨어"로 정의한다[DoD2009]. 가장 일반적으로 쓰이는 OSS 정의로는 자유 소프트웨어 기구(Free Software Foundation: FSF)와 오픈 소스 이니셔티브

1) 애자일(Agility)은 기존의 기술을 빠르게 개선 또는 확장할 수 있는 성질을 말한다. 본 고에서는 애자일 개발 방법론(Agile Software Development - 일정한 주기를 반복하여 끊임없이 프로토타입을 제작해 내는 개발 방법론, "빠르고 낭비 없이"라는 애자일의 의미에서 이름을 가져왔다)과는 구분하여 사용한다.

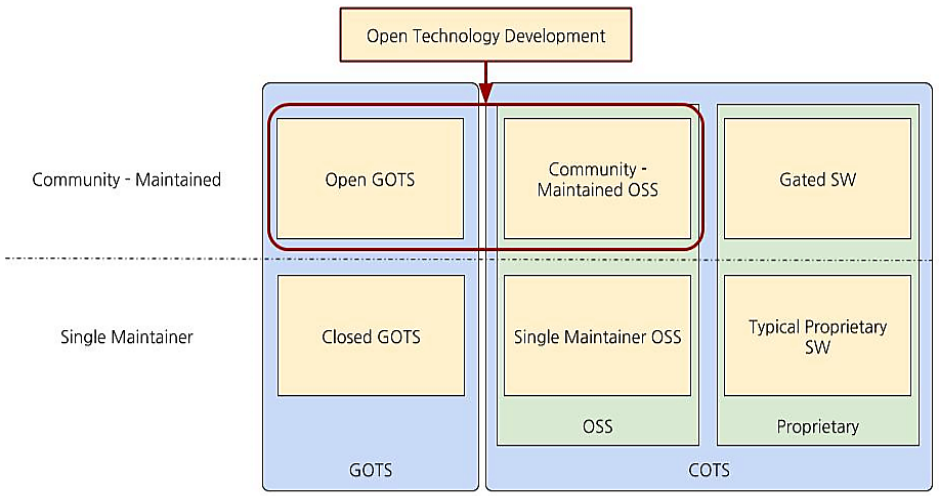
(Open Source Initiative: OSI) 정의가 있다.

OTD 개발에서 협업 효과를 높이기 위해 기성품(Off-the-shelf: OTS) 컴포넌트를 사용하고, 개발된 컴포넌트 역시 기성품으로 제작한다. 이 중 일부 OTS의 경우 미국 정부 내에서만 존재하는데, 이러한 OTS를 정부 OTS(Government OTS: GOTS)라고 부른다. 상업용 품목인 기성품은 상업용 OTS(Commercial OTS: COTS)이다. 개방형 GOTS(Open GOTS) 프로젝트는 OSS와 유사한 방식으로 소프트웨어를 개발/유지하기 위해 다기관 협력 개발 방식을 사용하는 GOTS 프로젝트이다. OSS는 하나 이상의 비정부 목적으로 대중에게 공개될 수 있으므로, U.S. Code Title 41, Chapter 7,section 403 & 431에 따라 상업용 소프트웨어가 될 수 있다. [그림 1]을 통해 GOTS, OGOTS, COTS와 OSS의 관계를 확인할 수 있다.

OSS와 GOTS의 차이는 다음 두 가지의 요인 때문에 발생한다.

- 정부는 개발 결과물을 더 개방할 수 있는 지적 권리를 가지고 있지 않다[2)]
- 정부는 잠재적인 적국(Potential Adversaries)에 소프트웨어를 제공하지 않음으로써 국가 안보상의 이점을 유지하고자 한다

OTD 개발에서 GOTS 프로젝트는 COTS로 전환할 시기, 특히 OSS 정비로 전환하는 것을 잘 검토해야 한다. 프로젝트를 계속 GOTS로 유지하는 경우, 비슷한 기능을 하는 외부 솔루션이 등장함



〈자료〉 Scott, Wheeler, Lucas, &Herz, "Open Technology Development(OTD): Lessons Learned and Best Practices for Military Software," 2011, p.5.

[그림 1] 기성품(OTS) 유지보수 전략

2) 예; 정부는 정부 목적 권리(Government Purpose Rights: GPR)를 가지고 있지만 무제한의 권리는 가지고 있지 않다.

에 따라 해당 컴포넌트가 고립될 가능성이 있다.

2. 개방형 기술 개발 과정

OTD 개발은 다음의 여섯 단계(가능한 경우 7단계 포함)를 반복하면서 발전하는 과정을 거친다.

- 단계 1: 관련 기능을 가진 기존 OSS 프로젝트를 검색한다.
- 단계 2: 어떤 새로운 프로젝트가 필요한지, 어떤 기존 프로젝트가 OTD로 전환되어야 하는지를 확인한다.
- 단계 3: 각각의 프로젝트가 협업에 적법한 명확하고 단순한 라이선스를 가지고 있는지를 확인한다.
- 단계 4: OTD 사용 프로젝트는 관리(Governance)되어야 한다. 또한, 기여자(Contributor)에게 적정 수준의 보호를 제공하기 위해 OTD는 반드시 포크 가능해야 한다.
- 단계 5: 협력관계를 구축한다.
- 단계 6: 각 프로젝트에 대해 구현 요소, 재사용 요소, 구현 방법, 환경 등 주요 기술적 지침을 정한다.
- 단계 7: 프로젝트가 발표 가능하거나 주요 릴리즈 발생 같은 중요한 이벤트가 발생할 경우, 필요한 사람들에게 알려야 한다.

협업은 OTD의 핵심 부분이기 때문에, OTD 문서에서는 협업에 대한 많은 지침을 제공한다. 문서에서 OTD 프로젝트는 프로젝트 사이트와 함께 협업에 필요한 인프라를 반드시 구축해야 한다고 기술하고 있다. 이때 중앙 프로젝트 사이트(Central Project Site)는 다음과 같은 기능을 가져야 한다.

- 프로젝트에 관심 있는 사람들에게 출발점과 관련 정보를 제공한다.
- 오류사항을 보고하거나 새로운 특성을 요청할 수 있는 메커니즘을 제공한다.
- 소프트웨어 구성 관리(Software Configuration Management: SCM) 메커니즘을 제공한다. 이는 변화를 누가, 언제, 어떻게 진행했는지를 기록하고 열람할 수 있도록 해주는 시스템이다.
- 유저와 개발자 간 여러 이슈를 상의할 수 있는 창구를 제공한다.
- 주요 배포 저작물을 내려 받을 수 있도록 한다.

협업 시 차질 없는 진행을 위해 원활한 커뮤니케이션이 필요하다. 협업 인프라와 커뮤니케이션 관련한 자세한 사항은 OTD 문서에서 확인할 수 있다.

DoD Instruction 5000.02 문서[10]에서는 OTD 프로젝트를 새로 수립하거나 기존의 프로젝트를 전환할 때에 대한 배경을 설명하고 있다. 이 경우 대안 분석(Analysis of Alternatives: AoA), 정보 요청(Request for Information: RFI), 제안 요청(Request for Proposal), 그리고 제안서 평가(Evaluating Proposals) 과정을 필요로 한다. 각 과정에 대한 자세한 내용은 OTD 문서 3절에서 확인할 수 있다.

3. 개방형 기술 개발의 장점

OTD는 속도와 비용, 혁신 측면에서 많은 장점이 있다. 소스 코드가 공무원과 계약업체 모두에게 공개되어 있으므로 기존의 소스 코드를 활용한 솔루션을 만들 수 있다. 또한, 이미 대금을 지불한 작업물을 재사용함으로써 빠르게 시스템을 변화시킬 수 있다. 이에 따라 개발자는 오로지 기존의 기능 변화나 병합에만 집중할 수 있으므로 새로운 기능 개발을 위한 시간을 줄일 수 있다. 이는 더 효율적인 인력 사용으로 인한 혁신 증가의 효과도 가져온다. 독점 임대료의 감소와 더불어 경쟁 체제로 인한 결과물의 질적 향상 역시 가질 수 있다.

정보 보증 및 보안에서도 강점이 있다. 일반적으로 사람들은 소스 코드를 숨기는 것이 보안에 도움이 된다고 생각한다. 하지만 소스 코드를 숨기는 것은 공격에 대한 충분한 대응이 아니다. 이러한 문제에 대해 DoD에서는 다음과 같이 설명하고 있다.

> *"동적 공격은 소스나 바이너리가 필요하지 않다. ... 소스 코드가 필요하더라도, 적절한 소스 코드를 ... 에 의해 다시 생성하여 취약성을 검색할 수 있는 경우가 많다. ... 일반인이 사용할 수 있는 소스 코드를 만드는 것은 공격자만이 아니라 수비 측에도 큰 도움이 된다. ... 소스 코드를 숨기는 것은 취약성에 대한 제 3자의 대응 능력을 억제하지만, 이것은 분명히 보안상의 이점이 아니다(Frequently Asked Questions regarding Open Source Software(OSS) and the Department of Defense (DoD))"*

물론 OSS가 독점적 소프트웨어보다 보안 측면에서 훨씬 더 강하다는 것은 아니다[11]. 그러나 다양한 사람들에게 코드를 보임으로써 취약점을 더 빠르게 발견할 수 있고, 이에 더 빠르게 대응할 수 있다는 점에서 확실한 장점을 가진다고 할 수 있다.

III. 미국의 OSS 사용 지침 및 관련 법률

1. OSS 사용 권고에 대한 지침

관리예산처(Office of Management and Budget: OMB)에서는 M-04-16 메모(Memorandum)[12]에서 소프트웨어 취득에 관한 기존 연방 정책이 독점적(Proprietary) 소프트웨어 및 OSS 모두에 동일하게 적용된다고 기술한다. 또한, DoD CIO(Chief Information Officer)의 "OSS에 대한 지침의 명확화(2009.10.)" 메모[10]에서는 DoD용 소프트웨어에 대한 시장조사를 실시할 때 고려해야 할 OSS의 긍정적인 측면을 명시하고 있다.

미 정부의 OSS에 대한 일반 정책과 지침 문서는 미국 정부 내에서 OSS가 사용될 수 있음을 명확히 나타낸다. 특정 OSS 제품이 적합하지 않을 수는 있지만, OSS 제품 사용 자체는 미 연방 및 DoD 규칙 하에서는 문제가 되지 않는다.

2. OSS 라이선스와 공개에 관한 법률

OTD 개발 결과물의 개방 여부는 다음과 같은 두 가지로 나타낼 수 있다. 첫째, 정부가 OSS 라이선스 하에 결과물을 공개하는 경우이다. 미국 정부 관계자에 의해 개발된 소프트웨어는 17 U.S.C. 105 조항[13]에 따라 저작권에 구애받지 않는다. 이러한 소프트웨어는 저작권 보호를 받을 수 없기 때문에 흔히 'Public Domain'이[3] 된다. 만약, 개발 품목이 기존 OSS 프로젝트의 수정 혹은 확장 결과물인 경우, 기존 OSS 프로젝트의 라이선스를 따르거나 저작권 보호가 없는 Public Domain 라이선스로 소프트웨어를 출시할 수 있다. 이 경우 기존의 프로젝트와 통합할 수 있다.

수정 사항에 저작권 보호가 존재하는 경우에 대해 OTD 문서는 듀얼 라이선스(Dual License)[4] 하에서 소프트웨어를 배포하는 것을 추천한다. 이 경우 잘 쓰이는 라이선스와 잘 쓰이지 않는 라이선스 두 개의 라이선스 하에서 두 개의 소프트웨어를 동시에 배포한다. 이를 통해 해당 프로젝트가 공용 OSS 라이선스로 보다 쉽게 전환될 수 있다.

하청업체의 경우 특정 상황에서 소프트웨어를 OSS로 배포할 수 있다. 정부가 계약자에게 저작권을 주장할 권리를 부여한 경우, 그리고 정부의 자금을 이용하여 독자적으로 개발한 경우가 그러하다.

3) Public Domain은 어떠한 작업물이 저작권에 전혀 구애받지 않고 자유롭게 이용될 수 있을 때 쓰인다. Public Domain 하에서 그 누구도 해당 작업물을 소유하지 않는다.

4) 소프트웨어를 서로 다른 두 가지 라이선스 하에 각각 동시에 두 개 배포하는 것

미 정부는 많은 경우에서 저작권이 아니라 무제한적 권리(Unlimited Rights)를 가진다. 정부가 가진 무제한적 권리는 OSS로 소프트웨어를 공개할 때 저작권자와 동등한 권리를 가지게 한다. 정부는 무제한적 권리보다 하위의 권리를 가지는 것을 주의한다. 이는 소프트웨어 개발 시 계약자가 개발 품목에 의도적으로 독점적 의존성을 만들어 경쟁 시스템을 무의미하게 바꿀 수 있기 때문이다.

OTD 프로젝트를 대중에 공개할 때에는 대외비 혹은 수출 제어(Export Control)에[5] 해당하는 품목인지를 역시 확인해야 한다. 국방 소프트웨어의 특성상 대외비 컴포넌트가 존재하는 품목이 있을 수 있으며, 이 경우 합법적으로 공개할 수 없다. 따라서 대외비 사항이 포함되는 소프트웨어의 경우 대외비 컴포넌트와 그렇지 않은 컴포넌트를 나누어 관리한다. 이렇게 되는 경우 대외비가 아닌 컴포넌트는 대중에 공개할 수 있다.

IV. 현재 미 국방 소프트웨어에서의 OSS 거버넌스 현황

DoD에서는 정부 내의 협력적 개발을 위해 여러 거버넌스 도구를 제공한다. DoD의 오픈소스 프로젝트는 GitHub, SourceForge, Forge.mil, Google Code 등에서 찾아볼 수 있다. 또한, DoD 내부에서 사용하는 거버넌스 도구로는 DISA(Defense Information Systems Agency)의 Forge.mil과 DDS(Defense Digital Service)의 Code.mil이 있다.

1. Forge.mil

DoD의 DISA는 Forge.mil 플랫폼을 통해 “협력적 개발과 더불어 오픈 소스와 커뮤니티 소스 소프트웨어 사용을 가능하게 한다[11]”. Forge.mil은 DoD에서의 기술 개발 커뮤니티를 위한 서비스를 제공한다. Forge.mil은 ProjectForge와 Software Forge로 이루어져 있다, ProjectForge는 프라이빗 개발 환경을 제공하고, SoftwareForge는 협력적 커뮤니티 개발 환경을 제공한다.

Forge.mil은 굉장히 빠르게 성장하고 있다. 2008년 처음 수립된 이후, 2011년 발표한 자료[14]에 따르면 프로젝트 수는 500개, 유저 수는 10,000명이었으나, 최근 브로슈어 자료[15]에 따르면 현재 유저는 45,000명, 프로젝트 수는 1,834개로 네 배에 가까운 성장세를 이루었다. 미국이 협력적 개발 방식을 권고함에 따라 Forge.mil은 더욱더 활성화될 것으로 보인다.

5) 수출 통제 규제(Export Control Regulation)는 허가받지 않은 상품이나 정보 등을 수출하는 것을 방지하는 정부 규약을 말한다.

2. Code.mil

DoD의 DDS에서는 2017년 소프트웨어 DoD 프로젝트 개발자 간 협업을 위해 Code.mil을 발표[16]하였다. 이는 정부 OSS를 위한 Code.gov를 DoD용으로 특화한 플랫폼으로, Code.mil에 공개된 프로젝트는 Code.gov에서도 확인할 수 있다.

Code.mil은 첫 번째 오픈소스 프로젝트인 eMCM을 배포함으로써 DoD의 소프트웨어 공개 전환을 시작하였다. 2018년 현재 eMCM을 포함한 6개의 저장소(Repository)를[6] code-mil 깃허브에서 확인할 수 있다.

V. 결론

OSS 개발 방법은 보안과 속도 면에서 여러 장점을 가지고 있다. Eric Raymond는 "보고 있는 눈이 충분히 많으면, 찾지 못할 버그는 없다[17]"고 말했다. 이는 OSS가 줄 수 있는 보안상의 대표적인 이점을 말하고 있다. 또한, 개발 속도 측면에서도 OSS는 큰 강점을 가지고 있다. 기존의 개발 프로세스에서는 "많은 기술적 노력과 아이디어들이 이렇다 할 성과를 내기 전에 수많은 회의와 브리핑을 거쳐야 한다[18]". OSS는 여러 단계의 브리핑과 회의를 '협업'이라는 이름 아래 하나로 통일시키며, 이에 따라 빠른 제작과 유지보수를 가능하게 한다. OTD는 이러한 OSS의 장점을 국방 프로젝트 개발에 접목시킨 개발방법이라고 생각한다.

OSS가 반드시 완벽한 해결점이라고 생각하는 것은 아니다. 경우에 따라서는 프로젝트를 공개하지 않는 것이 더 나은 선택이 될 수 있다. 하지만 프로젝트를 아예 공개하지 않고 독점 소프트웨어를 사용하는 것보다, 적재적소에 OSS를 사용하고 공개적 개발 방법을 사용하는 것은 최종 발주처인 사용자에게 더 넓은 선택의 폭을 제공할 수 있다. 이에 국내에서도 국방SW 무기체계 적용 가이드라인에 OSS 사용에 대한 OSS간 라이선스 충돌 관리, 사용한 OSS와 프로젝트간 라이선스 충돌 관리 등에 대한 고려와 OSS에 대한 주기적인 버전 관리를 통한 보안취약점에 대한 지속적인 관리의 추가가 필요해 보인다. 그리고 미국과 같이 OSS 장점을 적극 활용하기 위해서는 군내에 개방형 연구개발 방식에 대해 지속적인 연구 및 방법론 검토가 필요하다고 생각한다.

6) 총 8개의 저장소 중 아카이브된 2가지 저장소 제외

[참고문헌]

[1] 김영곤, “바이너리 파일에 기반한 오픈소스 취약점 분석 방안”, 인사이너리, 제6회 국방/항공 SW 기술 세미나, 2018. 3. 21.
[2] 김혜영 “오픈소스 보안 가이드라인 및 BDSK 오픈소스 보안 컨설팅 소개”, 블랙덕코리아 오픈소스 컨퍼런스 2018, 2018. 5. 24.
[3] 방위사업청, “무기체계 소프트웨어 개발 및 관리 매뉴얼”, 2018. 11. 2.
[4] 방위사업청, “공개SW 무기체계 적용 가이드라인”, 부록 14 - 2018. 11. 2.
[5] 전자신문, 국방부, “MS 종속 탈피 ‘개방형OS’ 확대 추진”, 2019. 3. 6.
[6] Scott, Wheeler, Lucas, & Herz, “Open Technology Development(OTD): Lessons Learned & Best Practices for Military Software”, Department of Defense, 2011. 5. 16.
[7] 월간SW중심사회, “국방분야 OS 및 상용 SW 사용실태와 문제점 조사”, 2017. 7. 7.
[8] 서영희, “국방 분야에서 공개SW활용 동향”, 소프트웨어정책연구소, 2018. 5. 31.
[9] Herz, Lucas, &Scott, “Open Technology Development: Roadmap Plan”. Department of Defense, 2006. 4.
[10] Department of Defense, “Operation of the Defense Acquisition System”, Department of Defense Instruction 5000.02, 2013. 11. 26.
[11] Nick Heath, “Six open source security myths debunked - and eight real challenges to consider”, ZDNet, 2013. 4. 23.
[12] Office of Management and Budget, “Software Acquisition”, M-04-16, 2004. 7. 1.
[13] U.S. Code, “Subject matter of copyright: United States Government works”, Title 17, Chapter 1, Section 105.
[14] Martin, & Lippold, “Forge.mil: A Case Study for Utilizing Open Source Methodologies Inside of Government”, Forge.mil Community Management Team, 2011.
[15] DISA, “Cloud-based ALM for Application Development, Collaboration and Source Control”, forge-brochure
[16] Department of Defense, “DoD Announces the Launch of ‘Code.mil,’ an Experiment in Open Source”, News Release, NR-077-17, 2017. 2. 23.
[17] Raymond, “The Cathedral and the Bazaar”, 1999. 9. 30.
[18] Defense Digital Service, “Code.mil: An Open Source Initiative at the Pentagon”, 2017. 3. 13.