논문 14-39C-04-06
The Journal of Korea Information and Communications Society '14-04 Vol.39C No.04
http://dx.doi.org/10.7840/kics.2014.39C.4.362

제어프로토콜 퍼징 기반 열차제어시스템 취약점 검출 기법

김 우 년◆, 장 문 수*, 서 정 택**, 김 상 욱°

Vulnerability Discovery Method Based on Control Protocol Fuzzing for a Railway SCADA System

Woo-Nyon Kim◆, Moon-Su Jang*, Jeongtaek Seo**, Sangwook Kim°

요 약

열차전력원격감시제어시스템은 열차 운행에 필요한 전력을 공급하는 제어시스템이다. 열차전력원격감시제어시스템은 관제실에서 원격지의 RTU(Remote Terminal Unit) 장치에 제어명령을 내리면, RTU는 이를 해석하여 입출력 모듈을 제어하고, 각 입출력 모듈로부터 수집된 데이터는 사령실로 전송함으로써, 사령실에서 열차에 공급되는 전력 상태를 감시 및 제어하는 시스템이다. 본 논문은 열차전력원격감시제어시스템의 구성요소인 VxWorks 실시간 운영체제 기반의 RTU에서 동작하는 DNP3 제어프로토콜 통신 소프트웨어의 취약점을 검출하기 위한 퍼징 기술에 대하여 제안한다. VxWorks는 동작중인 프로세스의 상태를 감시하기 위한 모니터링 에이전트를 설치할 수 없기 때문에, 모니터링 에이전트 없이 프로세스의 상태 정보를 획득하기 위해서 VxWorks의 디버그 채널을 이용하였다. 또한 DNP3 제어프로토콜은 제어명령을 순서대로 전송하지 않을 경우 패킷을 무시하기 때문에 제어프로토콜 함수 코드 순서를 고려한 퍼징을 수행하였다. 이러한 두 가지 기술을 이용하여 VxWorks 기반의 RTU에서 실행되는 DNP3 제어프로토콜 응용프로그램의 취약점을 시험할 수 있었다.

Key Words : Fuzzing, DNP3, Vulnerability Assessment, Control System Security, SCADA Security

ABSTRACT

A railway SCADA system is a control systems that provide the trains with the electricity. A railway SCADA system sends commands to the RTUs(remote terminal unit) and then it gathers status information of the field devices in the RTUs or controls field devices connected with the RTUs . The RTU can controls input output modules directly, gathers the status information of the field devices connected with it, and send the information to the control center. In this way, a railway SCADA system monitors and controls the electricity power for running trains. The cyber attackers may use some vulnerabilities in the railway SCADA system software to attack critical infrastructures. The vulnerabilities might be created in the railway software development process. Therefore it need to detect and remove the vulnerabilities in the control system. In this paper we propose a new control protocol fuzzing method to detect the vulnerabilities in the DNP3 protocol based application running on VxWorks in RTU(Remote Terminal Unit) that is a component of the centralized traffic control system for railway. Debug-channel based fuzzing method is required to obtain process status information from the VxWorks

◆ First Author : The Attached Institute of ETRI, wnkim@ensec.re.kr, 정회원
° Corresponding Author : Kyungpook National University, kimsw@knu.ac.kr, 정회원
* ETRI 부설연구소, moonsujang@ensec.re.kr, 정회원, ** ETRI 부설연구소, seojt@ensec.re.kr
논문번호 : KICS2014-01-009, Received January 15, 2014; Revised April 7, 2014; Accepted April 7, 2014

Ⅰ. 서 론

제어시스템은 생산의 자동화가 이루어진 공장을 비롯한 교통 관리, 전력 생산, 자원 관리 등 국가 기반 시설 전반에 걸쳐 널리 사용되고 있다[1]. 제어시스템은 기반시설을 안전하고 신뢰성 있게 운영하기 위해 원격의 정보를 수집하고 제어를 수행하는 시스템이다. 따라서 제어시스템 개발과정에서 만들어진 버그는 기반시설이 사이버 공격을 받게 되는 취약점으로 악용될 수 있다[2]. 제어시스템의 취약점은 기존 IT 시스템의 취약점과 달리 물리적인 장치를 제어하는 제어기기를 공격함으로써 물리적 피해를 유발할 수 있기 때문에 그 위험성이 매우 높다. 즉, 제어시스템 공격자나 악성코드는 네트워크, 운영체제, IT 응용 등의 IT 계층을 통해 감염 및 전파하여, 제어시스템 계층의 제어기기 및 컨트롤러를 불법 제어함으로써, 물리적 계층의 장치들이 오동작을 일으킬 수 있다[3]. 이러한 특성으로 인하여 제어시스템 공격에 의해 발생하는 피해는 대규모 인명피해를 유발하고 국가 이미지를 실추시킬 수 있다[4]. 따라서 악의적인 공격자가 제어시스템에 내재된 취약점을 찾아 사이버-물리 공격을 수행할 수 없도록 제어시스템의 취약점을 조기에 검출하여 대책을 수립하는 것이 매우 중요하다.

소프트웨어의 취약점을 검출하기 위한 연구로는 정적분석, 퍼징, 침투테스트 등의 방법이 있다[5]. 정적분석은 프로그램을 실행하지 않고 프로그램 텍스트를 정적으로 분석하여 프로그램 내의 취약 요소를 찾아내는 방법으로써 사람에 의한 매뉴얼 분석이나 정적분석 도구를 이용하는 분석 방법이 있다[5]. 수백만 라인의 소스코드를 짧은 시간에 분석하기 위해서는 보안약점과 관련된 패턴 등 진단규칙이 반영된 정적분석 도구를 이용하며, 이러한 분석 도구는 진단규칙 보유 수준과 이를 기반으로 한 진단능력 등이 성능에 중요한 영향을 미친다[6]. 정적분석은 프로그램을 실행하기 위한 환경구축이 필요 없으며, 개발 과정에서 공통적으로 나타나는 소프트웨어 버그를 제거하는데 효과적인 반면, 높은 오탐율과 운영환경 및 설정 상에서 발생하는 취약점을 분석하기 어려우며, 특히 프로그램 소스가 확보되어야 하는 단점이 있다[5]. 퍼징 테스트는 랜덤한 입력 스트링을 만들어 시험하는 방법[7]과, 파일, 네트워크 프로토콜, API 함수 등의 대상에 대해 비정상적인 입력을 주입하여 다양한 경계영역을 정상처리 하는지를 시험하는 자동화된 방법[8] 등이 있다. 퍼징 테스트는 이해하기 쉽고 단순하며 오탐이 없으며 자동화하기 쉬운 장점이 있지만, 높은 랜덤성으로 인해 미탐율이 높고 장시간 시험이 소요되는 단점이 있다[5]. 침투 테스트는 악의적인 공격자 입장에서 시뮬레이션 된 공격을 수행하여 시스템의 보안 상태를 평가하는 기술로써, 시험 대상의 사전지식 여부에 따라 블랙박스기반 테스트, 화이트박스기반 테스트, 그레이박스기반 테스트로 구분할 수 있다[5]. 이 기술은 오탐이 없고 실제 운영환경을 기반으로 시험을 수행함으로써 다른 도구로는 찾을 수 없는 취약점을 발굴할 수 있는 장점이 있지만, 공격자의 기술 의존성이 높고 분석 대상 시스템에 미치는 영향이 높은 단점이 있다[5]. 따라서 제어시스템 소프트웨어 소스의 확보가 어렵고 침투테스트 전문가 없이도 제어시스템 구성요소인 제어기기나 컨트롤러의 취약점을 검출하기 위해서는 퍼징 기술이 적합하다. 특히 Modbus, DNP3(Distributed Network Protocol)와 같은 제어시스템에 특화된 별도의 제어프로토콜로 개발된 제어기기의 취약점을 검출하기 위해서는 제어프로토콜 특성을 반영한 퍼징 기술이 필요하다.

본 논문은 열차제어시스템의 하나인 열차전력원격감시제어시스템의 구성요소인 VxWorks 실시간 운영체제 기반의 RTU(Remote Terminal Unit)에서 동작하는 DNP3 제어프로토콜 통신 소프트웨어의 취약점을 검출하기 위한 퍼징 기술에 대하여 제안한다. VxWorks 기반의 RTU에는 퍼징 상태를 모니터링 하는 에이전트를 설치할 수 없으며, DNP3 제어프로토콜의 메시지 순서를 고려하지 않을 경우 수신측에서 패킷들을 버리게 되므로 원하는 퍼징 결과를 획득할 수 없다. 따라서 에이전트 설치 문제를 해결하기 위하여 VxWorks의 디버그 채널을 이용한 통신 프로세스 상태 모니터링 기법과 DNP3 제어프로토콜의 함수 코드의 실행 순서를 고려하여 퍼징 하는 기법을 개발하였다. 이러한 두 가지 기법을 적용하여 VxWorks 기반의 RTU에서 실행되는 DNP3 제어프로토콜 통신 소프트웨어에 대한 퍼징을 통해 취약점을 사전에 검출할 수 있다.

본 논문은 총 6장으로 구성되어 있다. Ⅱ장에는 열차제어시스템의 구조와 제어프로토콜, 그 중에서도 열차전력원격감시제어시스템에 사용하는 DNP3 제어프로토콜에 대해서 설명한다. Ⅲ장에서는 관련연구로써 제어프로토콜 퍼징 기술에 대해서 설명하고, Ⅳ장에서는 제어프로토콜 퍼징 기반 제어기기 취약점 검출 기술에 대해서 설명한다. Ⅴ장에서는 DNP3 Fuzzer의 설계와 구현에 대해서 설명 후 다른 퍼저와 비교를 통해 제시한 퍼저의 우수성에 대해서 비교 설명하고, Ⅵ장에서 이 논문의 결론과 향후 연구방향을 제시한다.

Ⅱ. 열차제어시스템과 제어프로토콜

2.1 열차제어시스템

제어시스템은 SCADA(Supervisory Control And Data Acquisition) 시스템, DCS(Distributed Control System), PLC(Programmable Logic Controller)를 장착한 시스템을 통칭하는 것[9]으로써 미국에서는 산업제어시스템이라고도 한다. 이러한 제어시스템은 그림 1과 같이 제어센터, 현장제어장치, 그리고 제어센터와 현장제어장치를 연결하는 통신 영역으로 구성된다[9].

열차제어시스템은 신호제어시스템, 열차전력원격감시제어시스템, 설비제어시스템, 통신제어시스템이 있다. 신호제어시스템은 열차의 운행제어, 속도제어, 간격제어를 수행하고, 열차의 운행상황을 표출하여 예외적인 상황을 효율적으로 통제하는 시스템이다. 이를 위해서 관제 실에는 ATS(Automatic Train Supervision) 시스템이 설치되고, 각 역사에는 인터페이스 컴퓨터(I.F 장치), 지상ATP(Automatic Train Protection), 전자연동장치(Electronic Interlocking)가 설치되어 궤도회로, 선로전환기, 진로개통표시기 등의 물리적 장치의 상태를 수집 및 제어한다. 또한 ATS 시스템 및 지상ATP는 열차내의 차상ATO(Automatic Train Operation) 장치 및 차상ATP 장치와 무선 통신을 통해 열차 간 자동속도조절, 열차안전간격 조정 등의 기능을 수행하여 열차가 안전하게 운행되도록 제어한다[12]. 그림 2는 신호제어시스템의 구성이다. 설비제어시스템은 역사의 기계설비, 에스컬레이터, 엘리베이터, 환기설비, 배수설비, 소방설비 등을 감시 및 제어하는 시스템이다. 통신제어시스템은 종합사령실과 현장 간 모든 음성 및 데이터를 실시간으로 전송하는 시스템으로 디지털 전송설비, 열차무선설비, 행선안내게시, 화상 전송설비, 원격방송장치 등으로 구성되어 있다.

열차전력원격감시제어시스템은 열차 및 역사의 시설물에 공급되는 전력계통을 감시 및 제어하는 시스템으로써, 관제 실은 전력제어를 위한 호스트, MMI,

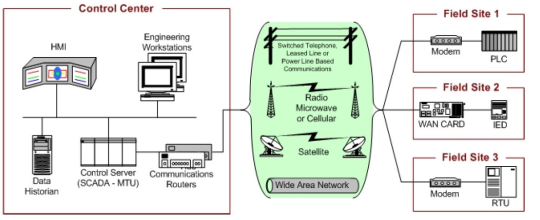

그림 1. 스카다 시스템의 일반적인 구성도[6]
Fig. 1. SCADA System Architecture

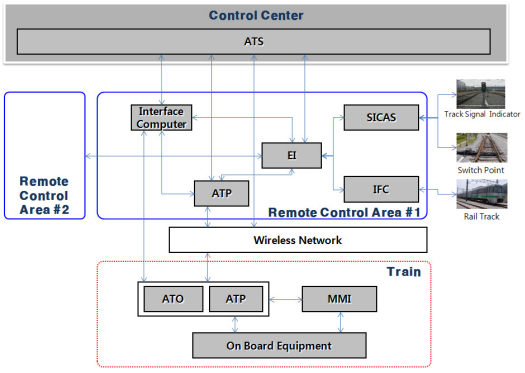

그림 2. 열차 신호제어시스템 구조
Fig. 2. C.T.C System Architecture

FEP 등의 설비가 위치하고, 역사에는 RTU가 설치되어 관제실로부터의 명령을 받아 정보를 수집하거나 제어를 수행한다. 그림 3은 열차전력원격감시제어시스템의 구조이다. 관제실의 FEP과 역사에 설치된 RTU는 Modbus, DNP3와 같은 제어프로토콜을 이용하여 통신을 수행한다.

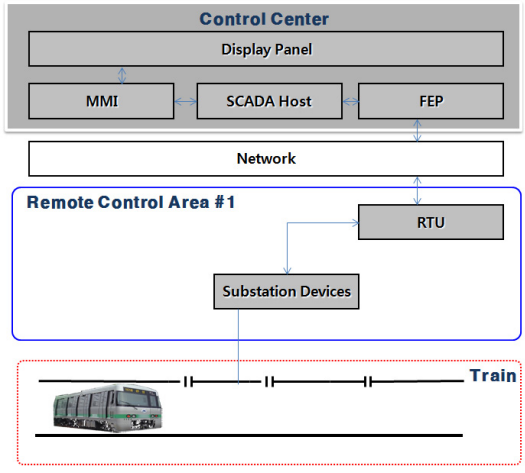

그림 3. 열차전력원격감시제어시스템 구조
Fig. 3. SCADA System Architecture

2.2 DNP3 제어프로토콜

본 절에서는 열차전력원격감시제어시스템에서 사용하는 DNP3 제어프로토콜의 동작 특성에 대해서 설명한다. DNP3 제어프로토콜은 국제전기위원회(IEC, International Electronical Commission)에서 제어시스템을 위해 정의한 IEC60870-5를 기반으로 개발된 제어프로토콜이다. DNP3 제어프로토콜은 제어센터의 DNP Master가 현장제어장치인 RTU의 DNP Slave로 주기적으로 요청 메시지를 전송하면 Slave는 요청 메

시지에 따라 해당하는 정보를 Master로 전송하게 된다. 이러한 유형의 통신은 현장의 모니터링 정보를 주기적으로 수집하는데 사용되며 그림 4와 같이 동작한다.

이외에도 사용자가 원하는 시점에 RTU에 상태정보 수집을 요청하거나, 제어명령을 요청하는 등의 Master가 발생시키는 이벤트에 의한 통신 방법, Slave에서 관리하는 물리적인 장치들이 발생한 이벤트를 설정에 따라 Slave가 자동으로 Master로 우선 전송하는 “Unsolicited Response” 통신 형태가 있다. 본 논문에서는 주기적 폴링 방법을 이용하여 RTU에 설치된 DNP Slave 프로그램의 취약점을 검출하는 기법에 대해서 설명한다.

DNP3 제어프로토콜의 계층구조는 데이터 링크 계층, 전송 계층, 응용 계층의 3계층으로 구성된다. 각 계층별 메시지 구조는 그림 5와 같이 구성된다.

사용자는 정보수집 및 제어를 원하는 객체 정보를 나타내는 Object Header와 처리 대상을 나타내는 Data로 구성된 메시지를 응용 계층으로 전달한다. 응용 계층은 메시지를 응용 헤더를 포함하여 2,048 바이트가 되도록 분할 후 응용 헤더를 추가하여 전송 계층으로 전달한다. 전송 계층에서는 분할된 메시지에 전송 계층 헤더를 추가하여 최대 255바이트 크기의 프레임으로 나누어 데이터 링크 계층으로 전달한다. 데이터 링크 계층은 프레임에 데이터 링크 헤더를 추가하고 CRC를 붙여 DNP Slave로 전송한다. DNP 프레임의 전송은 물리적 계층을 고려하여 18바이트에 맞추어 16바이트의 데이터와 2바이트의 CRC로 구성하

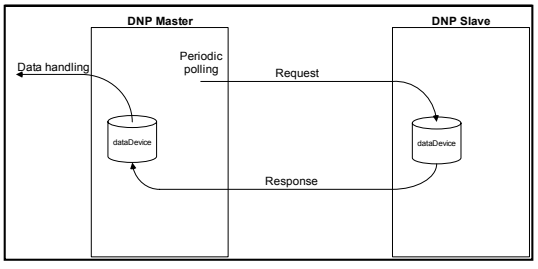

그림 4. 주기적 폴링에 의한 통신 방법
Fig. 4. Periodic Polling based DNP Communication

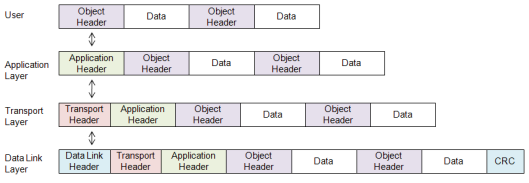

그림 5. DNP3 계층별 메시지 구조
Fig. 5. Layered Message Format of DNP3

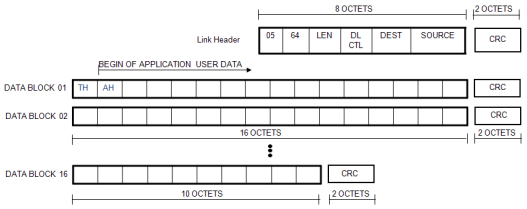

그림 6. DNP3의 프레임 전송 형식
Fig. 6. Frame Format of DNP3

여 그림 6과 같은 데이터 블록 형식으로 전송한다. 첫 번째 데이터 블록은 8바이트의 링크 헤더와 2바이트의 CRC로 구성된 10바이트가 전송되며, 두 번째부터 16바이트 데이터와 2바이트 CRC로 구성되어 데이터 블록이 전송된다. 마지막 데이터 블록은 남은 데이터 바이트 수와 2바이트의 CRC가 전달된다. 데이터 링크 계층에서 전송되는 프레임의 최대 크기는 255바이트의 데이터와 데이터 링크 계층 헤더 및 CRC를 합하여 최대 292바이트 크기로 한정된다.

Ⅲ. 제어프로토콜 퍼징

제어프로토콜 퍼징과 관련된 연구로는 TippingPoint사의 연구[10], KTH대학과 ABB에서 공동으로 수행한 연구[11], Dartmouth 대학의 LZFuzz 연구[2]가 있다.

TippingPoint사의 Devarajan은 DefCon에서 Sully 퍼징 프레임워크를 이용하여 Modbus, DNP3, ICCP 제어프로토콜에 대한 퍼징 기술을 제시하였다[10]. 해당 발표는 제어프로토콜에 대한 퍼징 기술을 최초로 제시한 것으로써 의미를 갖지만, 제어프로토콜 자체의 동작 특성을 고려하지 않고 단순히 블록 단위의 퍼징만을 수행하였다. 또한, VxWorks 기반의 현장제어장치를 대상으로 수행할 경우에는 테스트 케이스 입력에 의한 프로세스의 상태를 분석하고 프로세스에 문제가 발생할 경우 다시 실행시켜 주는 역할을 수행하는 모니터링 에이전트를 설치할 수 없다. 따라서 제어프로토콜 동작 특성을 반영하고, 퍼징 에이전트를 설치할 수 없는 문제를 해결해야만 VxWorks 기반 RTU에 대한 퍼징이 가능하다.

스웨덴의 KTH 대학에서는 제어시스템 제조사인 ABB와 공동으로 DNP3 프로토콜 구현상의 취약점을 시험하였다. 연구에서는 Flood 테스트, 취약점 스캐너를 이용한 스캐닝, 퍼징, 수동분석 등의 네 가지 분석 방법을 적용하여 분석하였으나 ABB와 체결한 비공개 협약에 따라 세부적인 시험내용 및 결과가 공개되지

않았다[11]. 다만 해당 연구에서도 Sully 퍼징 프레임워크를 이용해서 DNP3에 대한 테스트케이스 정의, 시험 대상 선정, 모니터링 에이전트 실행, 테스트 케이스 실행 등의 과정을 통해 결과를 취합하여 분석[11]하였기 때문에 해당 연구 역시 TippingPoint사의 연구와 동일한 한계가 있다.

Dartmouth 대학의 LZFuzz 연구는 제어시스템의 특성상 현장제어장치는 계속 운영되어야 하기 때문에 이를 모니터링하고 오류 발생 시 자동 재실행해 주는 Watchdog이 있으므로 별도의 에이전트를 설치하지 않는 방식으로 접근하였다[2]. 또한, 기존의 제어프로토콜 퍼징 연구와 달리 Master와 Slave 사이에 LZFuzz가 설치되어 Master에서 전송되는 패킷에 임의의 랜덤 값을 주입하여 퍼징 패킷을 생성 후 Slave로 전송하고, Slave의 응답을 LZFuzz가 수진하여 문제여부를 판단 후 Master로 전송하는 방식으로 동작한다[2]. 이러한 접근법은 퍼징 시스템이 Ⅱ의 2.2절에 설명된 DNP3 프로토콜의 분할 전송 특징에 맞게 프레임을 재조합 후 퍼징 정보를 삽입하고 다시 프레임을 분할하여 전송하는 기능을 수행해야 한다. 그러나 LZFuzz 연구는 제어프로토콜을 대상으로 실제 실험을 수행하지 않았기 때문에 제어프로토콜 퍼징에 적합한지를 파악할 수 없다. 또한, 시험을 위해서는 Slave 시스템뿐만 아니라 Master 시스템이 추가로 필요하고, Slave의 모든 기능을 시험할 만큼 충분한 동일 명령을 Master가 내릴 수 있는 환경 구축 등이 추가로 요구된다. 일반적으로 제어명령은 정해진 주기별로 특정 명령만을 주로 요청하기 때문에 잘 사용하지 않는 명령의 시험을 위한 별도의 방안도 요구된다.

현장제어장치에 별도의 에이전트를 설치하지 않고, DNP3의 통신 특성을 고려하여 제어프로토콜 응용프로그램에 대한 신규 취약점을 검출할 수 있는 기술이 필요하다.

Ⅳ. 제어프로토콜 퍼징 기반 제어기기 취약점 검출 기법

제어시스템의 현장제어장치인 RTU는 다양한 운영체제를 기반으로 개발될 수 있지만 본 논문에서는 열차전력원격감시제어시스템에서 일반적으로 많이 활용하고 있는 VxWorks RTU에서 실행되는 DNP3 제어프로토콜 통신 소프트웨어(DNP Slave)의 취약점을 검출하는 기법에 대해서 설명한다. RTU에 대한 취약점을 검출하기 위해서는 Ⅲ장에서 설명한 것처럼 다음의 두 가지 문제를 해결해야 한다. (1) DNP3의 통신 특성을 고려하여 퍼징을 수행해야 한다. 그렇지 않을 경우 전송된 퍼징 패킷은 RTU의 통신 프로그램에 의해 버려지게 되어 올바른 퍼징의 결과를 확보할 수 없다. (2) VxWorks 기반 RTU에는 퍼징 결과를 모니터링하고 퍼징 대상 프로그램 오류 발생 시점에 재시작 처리를 수행하는 모니터링 에이전트 프로그램을 설치할 수 없다. 따라서 별도의 모니터링 방안이 없다면 퍼징의 결과로 프로그램에 문제가 발생했는지를 확인할 수 없고, 어떠한 입력에 의해서 문제가 발생한지를 파악할 수 없다.

본 논문에서는 제어프로토콜 통신 특성을 고려하기 위하여 제어프로토콜 함수코드순서 기반 퍼징 기법과 디버그 채널 기반 퍼징 기법을 제안한다.

4.1 제어프로토콜 함수코드순서 기반 퍼징

DNP3 제어프로토콜을 이용하여 Master가 원격의 Slave로부터 상태정보를 읽고 쓰거나, Slave에게 제어를 요청하는 경우 등의 동작은 DNP3 통신의 특성상 일정한 순서에 따라서 진행이 되어야 한다. 예를 들면 상태정보를 읽어오기 위해서는 (1) 데이터 링크 계층에서 Reset Link 제어 함수 코드(0x00)를 설정한 데이터 블록을 전송하여 네트워크 연결을 설정하고, (2) 응용 계층에서 Read 함수 코드(0x01)를 설정한 메시지를 전송하는 두 단계의 과정을 거쳐서 Slave로부터 상태정보를 획득한다. 또한, 현장제어장치에 제어명령을 내릴 경우에는 (1) 응용 계층에서 Select 함수 코드(0x03)를 설정한 메시지를 전송하고, (2) 응용 계층에서 Operate 함수 코드(0x04)를 설정한 메시지를 전송하는 순서로 제어를 수행한다. 따라서 DNP3 제어프로토콜 퍼징을 수행하기 위해서는 하나의 메시지나 데이터 블록을 전달하는 것으로는 제대로 된 퍼징 결과를 확보하기가 어렵다. DNP3 제어프로토콜 퍼징은 설명한 것과 같이 함수코드의 순서를 반영하여 그림 7과 같은 시나리오 형태로 수행되어야 제대로 된 퍼징 결과를 확보할 수 있다. 또한, 매번 응용 계층의 함수코드에 대한 퍼징을 수행할 경우 Slave의 오류가 발생하여 네트워크 연결에 문제가 발생할 수 있으므로 항

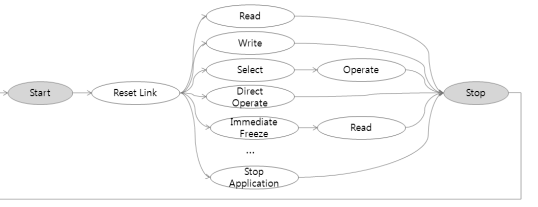

그림 7. 함수코드순서 기반 제어프로토콜 퍼징
Fig. 7. Function Code Ordering based Fuzzing

상 데이터 링크 계층의 Reset Link 제어 함수 코드를 이용하여 네트워크 연결을 재설정 후 퍼징을 수행해야 한다.

4.2 디버그 채널 기반 퍼징

일반적으로 퍼징 기술은 퍼징 패킷을 생성하는 퍼징 패킷 생성부와 퍼징 대상 시스템의 오류발생 여부를 확인하는 모니터링 에이전트로 구성된다. 모니터링 에이전트는 (1) 퍼징 패킷에 의한 대상 프로그램의 오류 발생 여부 확인, (2) 대상 프로그램의 비정상 종료 시 자동 재시작을 지원하여 퍼징 자동화 지원, (3) 오류 발생 시 메모리 및 네트워크 정보 수집 등의 역할을 수행한다.

VxWorks 기반 RTU에는 퍼징 결과를 모니터링하고 퍼징 대상 프로그램 오류 발생 시에 재시작 처리를 수행할 수 있는 모니터링 에이전트 프로그램을 설치할 수 없다. 그러나 일반적으로 VxWorks 기반의 장치는 프로그램 실행을 모니터링하기 위해서 디버그 채널을 제공하며, 이 채널을 통해서 프로그램의 오류 발생 여부를 확인할 수 있다. 본 논문에서는 VxWorks 기반 RTU에서 제공하는 디버그 채널을 이용하여 주입되는 퍼징 패킷에 의해 DNP3 Slave 프로그램에 오류가 발생하였는지를 모니터링 한다. 다음으로 퍼징 자동화를 위한 DNP3 Slave 프로그램 자동실행 문제는 제어시스템의 특성상 Watchdog이 프로그램의 문제를 감지하여 자동으로 실행해주기 때문에 해결 가능하다. 오류 발생 시 메모리 및 네트워크 정보 수집 역할은 오류 발생을 확인한 시점에 사용자가 재연을 통해 확인하도록 함으로써 퍼징을 수행할 수 있다.

V. 구현 및 평가

본 장에서는 DNP3 제어프로토콜 Slave 프로그램이 설치된 RTU를 퍼징하기 위한 도구인 DNP3 Fuzzer의 설계 및 구현에 대해서 설명하고, 다른 도구와의 차이점에 대해서 평가한다.

5.1 DNP3 Fuzzer 설계 및 구현

DNP3 Fuzzer는 윈도우 기반 시스템에 설치되어 운영되며 VxWorks 기반 DNP3 Slave 프로그램이 실행되는 RTU의 구현 취약점을 점검하는 도구이다. DNP3 Fuzzer는 퍼징 패킷을 생성하는 네트워크 퍼징 모듈, RTU의 Slave 프로그램 오류 발생을 모니터링하는 정보수집 모듈, 요청/응답 정보와 모니터링 정보를 관리하는 로그 모듈, 이들 정보를 취합하여 퍼징 결과를 분석하는 분석 모듈로 구성된다. 그림 8은 DNP3 Fuzzer의 구조이다.

네트워크 퍼징 모듈에서는 DNP3에 맞는 데이터 모델을 이용하여 함수순서에 따라 퍼징을 수행하기 위한 패킷을 생성하여 로그 모듈에 저장하고 RTU의 Slave 포트로 전송한다. 디버그 포트에 연결된 정보수집모듈은 VxWorks 쉘의 오류 정보 등을 수집하여 로그 모듈에 저장하면 분석 모듈은 오류 발생 여부를 판단 후 다음 퍼징 패킷을 생성하여 전송하도록 네트워크 퍼징 모듈에 정보를 제공한다. DNP3 Fuzzer 개발을 위해서 Sisco의 DNP 소스코드라이브러리를 이용하여 DNP3 패킷을 생성한 후 퍼징 정보를 삽입하였다. 그림 9는 DNP3 Fuzzer의 실행화면으로써, 좌측 트리 부분은 DNP3가 제공하는 함수코드를 나타내며, 체크된 순서대로 패킷을 전송하면서 가장 마지막 체크 부분의 함수코드를 퍼징 하도록 동작함으로써 함수코드순서 기반 퍼징이 가능하도록 하였다.

우측 상단은 RTU와 주고받는 통신 트래픽 패킷의 정보를 순서대로 표시하고 우측 하단은 DNP3 패킷을 프로토콜 구조에 따라 파싱한 정보를 제공한다. RTU

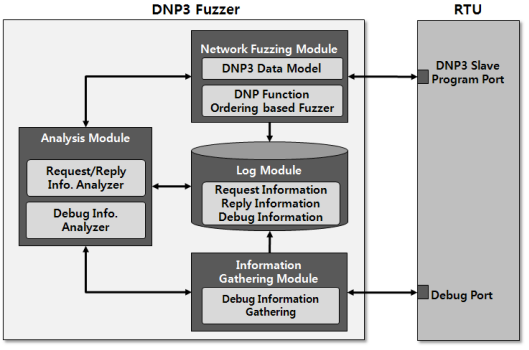

그림 8. 제어프로토콜 DNP3 Fuzzer 구조
Fig. 8. Design of DNP3 Fuzzer

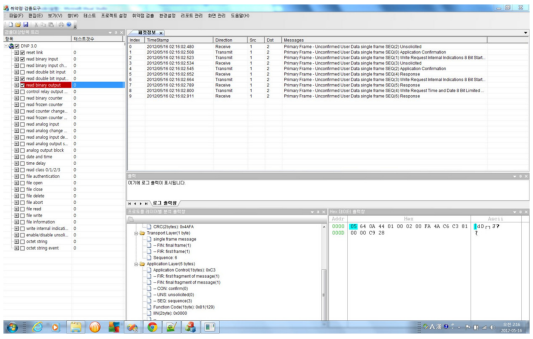

그림 9. DNP3 Fuzzer 실행 화면
Fig. 9. DNP3 Fuzzer Runtime Screen Shot

의 디버그 채널로부터 획득된 정보는 별도로 관리하며 오류가 발생한 정보를 상태 바에 출력하도록 구성하였다.

5.2 DNP3 Fuzzer 평가

본 절은 DNP3 Fuzzer를 기존의 연구와 비교 분석한다. 기존의 DNP3 제어프로토콜 퍼징 기술은 에이전트 설치가 가능한 시스템을 대상으로 하고, 패킷 단위의 퍼징을 수행하기 때문에 DNP3 제어프로토콜의 특징인 함수코드별로 생성되는 메시지 단위로 퍼징을 수행할 수 없다. 그러나 본 논문에서 제시하는 DNP3 Fuzzer는 VxWorks 실시간운영체제 기반의 현장제어장치인 RTU에 별도의 에이전트 설치 없이 퍼징을 수행하고 그 결과를 확인할 수 있다. 특히 DNP3 제어명령의 함수코드순서 기반으로 패킷 단위의 퍼징이 아니라 메시지 단위의 퍼징을 수행함으로써 RTU에 도착한 패킷의 버림 현상을 줄일 수 있도록 개발하였다.

표 1. DNP3 제어프로토콜 Fuzzer 비교
Table 1. Comparison DNP3 Fuzzer to Related Fuzzers

	DNP3 Fuzzer	TippingPoint DNP Fuzzer[10]	KTH Fuzzer[11]	LZFuzz[2]
DNP Fuzzing	O	O	O	X
Fuzzing Unit	DNP Function	Packet	Packet	Packet
VxWorks based RTU Fuzzing	O	X	X	X
Agentless	O	X	X	O
Monitoring Target	Process	Process Network	Process Network	-
Gathering Info.	Debug Shell	Process Network	Process Network	Network Response
Features	Standalone	Standalone	Standalone	Inline

Ⅵ. 결 론

본 논문은 열차제어시스템 중 열차에 전력 공급을 감시 및 제어하는 열차전력원격감시제어시스템의 구성요소인 VxWorks 실시간 운영체제 기반의 RTU (Remote Terminal Unit)에서 동작하는 DNP3 제어프로토콜 통신 소프트웨어에 내재된 취약점을 검출하기 위한 퍼징 기술에 대하여 제시하였다. VxWorks 실시간 운영체제의 디버그 채널을 이용하여 VxWorks에서 동작하는 프로세스의 상태 정보를 획득하는 디버그 채널 기반 퍼징 기법과 DNP3 제어프로토콜의 통신 특성을 고려하여 제어프로토콜 함수코드순서 기반 퍼징 기법에 대해서 설명하였다. 이러한 두 가지 기법을 이용하여 VxWorks 기반의 RTU에 대한 취약점 검출을 위한 퍼징 시험환경을 구축하였다. 개발된 도구를 활용하여 열차전력원격감시제어시스템에서 사용하는 RTU, FEP 등의 VxWorks 기반 제어기기에서 운영되는 제어프로토콜 응용의 취약점 여부를 사전에 시험하여 제거함으로써, 국가기반시설인 열차제어시스템이 사이버 공격으로부터 좀 더 안전하게 운영될 수 있는 기반을 마련할 수 있다.

퍼징 시험은 장시간에 걸쳐 시험을 수행해야 하며, 장시간에 걸친 시험에서 오류가 없다고 하더라도 프로그램에 취약점이 없다고 단언하기가 어렵다. 따라서 향후 연구과제로는 좀 더 짧은 시간에 높은 코드 커버리지를 확보할 수 있는 방안에 대한 연구를 통해 퍼징 시험의 수행시간을 줄이고 취약점을 검출하는 효율을 높이는 방안에 대한 연구가 필요하다.

References

[1] H. Yoo, J.-H. Yun, and T. Shon, "Whitelist-based anomaly detection for industrial control system security," *J. KICS*, vol. 38, no. 08, pp. 641-653, Aug. 2013.

[2] R. Shapiro, S. Bratus, E. Rogers, and S.W. Smith, "Identifying vulnerabilities in SCADA systems via fuzz-testing," in *Proc. Critical Infrastructure Protection*, pp. 57-72, Berlin Heidelberg, 2011.

[3] R. Langner, *To kill a centrifuge, a technical analysis of what stuxnet's creators tried to achieve*, The Langner Group, Nov. 2013.

[4] I. J. Kim, Y. J. Chung, J. Y. Koh, and D. Won, "A study on the security management for critical key infrastructure(SCADA)," *J. KICS*, vol. 30, no. 8C, pp. 838-848, Aug. 2005.

[5] B. Liu, L. Shi, Z. Cai, and M. Li, "Software vulnerability discovery techniques: A survey," in *Proc. Multimedia Inf. Netw. Security*, pp. 152-156, 2012.

[6] J. Bang and R. Ha, "Validation test codes development of static analysis tool for secure

software," *J. KICS*, vol. 38, no. 05, pp. 420-427, May 2013.
[7] B. P. Miller, L. Fredriksen, and B. So, "An empirical study of the reliability of UNIX utilities," *Commun. ACM*, vol. 33, 1990.
[8] P. Oehlert, "Violating assumptions with fuzzing," *IEEE Security and Privacy*, vol. 3, no. 2, 2005.
[9] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," NIST SP 800-82, pp. 83-98, Jun. 2011.
[10] G. Devarajan, "Unraveling SCADA protocols: Using sulley fuzzer," *Defon 15 Hacking Conf.*, 2007.
[11] A. B. M. Omar Faruk, "Testing & exploring vulnerabilities of the applications implementing DNP3 protocol," Master Thesis, KTH, 2008.
[12] S. Oh, Y. Yoon, M. Kim, and Y. Kim, "System configuration of communication network for Korean radio-based train control system," *Korea Soc. Precision Eng.*, pp. 989-990, 2012.

김 우 년 (Woo-Nyon Kim)
1996년 2월 : 안동대학교 컴퓨터공학과 공학학사
1998년 2월 : 경북대학교 컴퓨터과학과 이학석사
2000년 2월 : 경북대학교 컴퓨터과학과 박사수료
2000년 3월~2003년 12월 : (주) 니츠 선임연구원
2003년 12월~현재 : ETRI 부설연구소 책임연구원/실장
<관심분야> 기반시설보안, SCADA 보안, 제어시스템 보안, 취약점 분석, 네트워크 보안

장 문 수 (Moon-Su Jang)
2002년 2월 : 경성대학교 컴퓨터공학과 공학학사
2004년 2월 : 포항공과대학교 정보통신학과 공학석사
2005년 3월~현재 : ETRI 부설연구소 선임연구원
<관심분야> 침입탐지시스템, 침입감내시스템, 제어시스템 보안

서 정 택 (Jeongtaek Seo)
1999년 2월 : 충주대학교 컴퓨터공학과 공학학사
2001년 2월 : 아주대학교 컴퓨터공학과 석사
2006년 2월 : 고려대학교 정보보호대학원 정보보호공학 공학박사
2000년 11월~현재 : ETRI 부설연구소 책임연구원/부장
2011년 11월~현재 : 고려대학교 정보보호대학원 겸임교수
<관심분야> 스마트그리드 보안, 제어시스템 보안, 원자력 사이버보안, DDoS 공격 탐지 및 대응

김 상 욱 (Sangwook Kim)

1979년 2월 : 경북대학교 전자계산기공학 공학사
1981년 2월 : 서울대학교 컴퓨터과학과 이학석사
1989년 2월 : 서울대학교 컴퓨터과학과 이학박사
현재 : 경북대학교 IT대학 컴퓨터학부 교수
<관심분야> 모바일 멀티미디어 시스템, 멀티미디어 컨텐츠 저작 및 인간과 컴퓨터의 상호작용, 컴퓨터 보안