Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.1, Feb. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.1.179

산업제어시스템 취약성 분석을 위한 무작위성 기반 퍼징 테스트 케이스 평가 기법*

김 성 진,† 손 태 식‡
아주대학교

Randomness Based Fuzzing Test Case Evaluation for Vulnerability Analysis of Industrial Control System*

SungJin Kim,† Taeshik Shon‡
Ajou University

요 약

사물인터넷 시대의 도래로 인터넷에 연결되는 매체가 급격히 증가하고 있다. 사물인터넷은 삶의 편리성을 향상 시켰지만, 사생활 침해와 같은 보안 이슈를 야기하였다. 따라서 사물인터넷 시대의 시작인 현시점에서 가장 중점적으로 논의되어야 하는 부분은 사이버 보안이다. 특히 사물인터넷 제품 시장이 급격히 형성되고, 다양한 프로토콜이 혼재되어 사용됨으로써 생기는 보안 위협에 대응하기 위해 프로토콜에 빠르게 적용 할 수 있는 취약성 분석 방법이 필요한 상황이다. 본고에서는 분산과 엔트로피를 이용하는 새로운 무작위성 기반의 테스트 케이스 평가 방법론을 제안하여 사물인터넷 보안에 기여하고자 한다. 본고에서 제안하는 테스트 케이스 평가 방법은 기존 기법과는 달리 테스트 셋 크기에 상관없이 빠른 속도로 테스트 케이스의 평가를 수행할 수 있다.

ABSTRACT

The number of devices connect to the internet is rapidly increasing with the advent of the IoT(Internet of Things). The IoT has improved the convenience of life. However, it makes security issues such as privacy violations. Therefore cybersecurity is the most important issue to be discussed nowadays. Especially, various protocols are used for same purpose due to rapidly increase of IoT market. To deal with this security threat noble vulnerability analysis is needed. In this paper, we contribute to the IoT security by proposing a new randomness-based test case evaluation methodology using variance and entropy. The test case evaluation method proposed in this paper can evaluate the test cases at a high speed regardless of the test set size, unlike the traditional technique.

Keywords: Vulnerability Analysis, Fuzzing Test, Test Case Evaluation, Industrial Control System

I. 서 론

사물인터넷의 시대의 도입으로 인해 사용자의 편의성을 위해 다양한 제품들이 출시되어, 삶의 많은 영역에서 급격한 변화가 일어나고 있다. 불가 몇 년 전에는 인더스트리 4.0으로 인한 공장 자동화나 소수의 스마트 홈 제품들만 출시되었지만, 현재는 스마트홈, 스마트 팩토리를 넘어 도시 단위의 스마트 시티, 범국가적 단위인 스마트그리드에 이르기까지 다양한 영역으로 확대되고 있다. 이러한 사물인터넷

Received(12. 13. 2017), Modified(01. 29. 2018), Accepted(01. 29. 2018)
* 본 연구는 2016년도 산학협동재단의 학술연구 지원에 의하여 이루어진 연구로서, 관계부처에 감사드립니다.
† 주저자, ksjskyblue@ajou.ac.kr
‡ 교신저자, tsshon@ajou.ac.kr(Corresponding author)

영역의 확대와 함께 사물인터넷에 사용되는 사물 간 통신에 효율성 증가를 위해 새로운 프로토콜들이 개발되고 있다. 사물인터넷 전용 프로토콜들은 오버헤드를 줄이는데 초점을 맞추어 개발되어 효율적이지만, 새로운 프로토콜이기 때문에 이를 이용하는 제품들의 프로토콜 구현과정 중 실수로 생성된 구현상의 취약점들을 가지고 있을 가능성이 높다. 따라서 사물인터넷 프로토콜들의 구현상의 취약점 존재 여부에 대한 검증이 필요한 실정이다.

프로토콜 구현상의 취약성 분석 방법 중 가장 널리 활용되는 방법은 퍼징 테스트이다. 퍼징 테스트는 무작위로 생성된 값을 기본으로 하지만, 적은양의 테스트 케이스로 빠르게 코드 커버리지를 높이기 위한 방법들이 병행되어 사용되고 있다. 적은양의 테스트 케이스로 좋은 효과를 얻기 위해 테스트 케이스 평가에 대한 연구들이 이를 뒷받침하는데, 기존 연구들은 다수의 테스트 케이스를 평가하기에는 오랜 시간이 소모되는 단점이 존재한다.

따라서 본 논문에서는 효율적인 테스트 케이스 평가를 수행하고자 엔트로피와 분산을 이용하여 무작위성을 계산하고, 이를 바탕으로 테스트 케이스 평가를 수행한다. 테스트 케이스를 수행하기 위해 사용되는 인풋 값의 엔트로피와 분산 값을 계산하여 Shahbazi의 연구에서 테스트 케이스 평가의 측도로 사용한 거리와 동일하게 테스트 케이스의 평가에 사용한다. 이 평가 요소를 통해 테스트 케이스의 집합인 테스트 셋의 선택에 도움이 될 수 있다. 그리고 이 평가 특성은 다수의 테스트 케이스 평가에 적합하기 때문에 새로운 사물인터넷 프로토콜들 취약성 분석에 활용 가능할 것으로 사료되어 무작위성 기반의 블랙박스 퍼징 테스트 기법에 대해 논의한다.

본 논문의 2장에서는 최근 논의되고 있는 취약성 분석 기법에 대한 연구들을 분석하고, 그 문제점에 대해 논의한다. 이후 3장에서는 프로토콜 취약성 분석을 위한 테스트 케이스 평가 방법 개발을 위해 기존 테스트 케이스 평가 연구들을 분석과 프로토콜에 대한 고찰을 바탕으로 뛰어난 테스트 케이스를 생성하기 위한 조건을 도출한다. 이후 4장에서 이 조건을 만족하기 위한 테스트 케이스 평가 방법론을 제안하고, 이후 5장에서 이에 대한 검증을 수행한다. 마지막 6장에서는 제안 기법의 장점과 한계점을 토대로 활용 방안과 향후 연구에 대해 다룬다.

II. 취약성 분석 관련연구

퍼징테스트는 취약성을 검사하고 싶은 소프트웨어에 비정상적인 입력 값을 주입하여 비정상적인 상태를 유도하는 랜덤 테스트 방법으로 현재 효율성을 향상시키기 위한 다양한 연구들이 진행되고 있다[2]. 간단하게는 입력 값을 랜덤하게 생성하기 이전에 경계 값을 우선적으로 확인하는 기법이 Peng의 연구에서 사용되었다[11]. 이 연구를 바탕으로 본 연구진도 중첩된 구조(Nested Structure)를 가지는 프로토콜의 취약성 분석 방법에 대해 연구한 바 있다[7]. 이러한 연구들은 구현상 발생 가능한 취약점을 찾는데 용이한 방법으로, 기존 랜덤테스트의 단점을 극복하는 방법이 될 것으로 판단된다.

이러한 퍼징테스트의 단점을 극복하는 연구들은 이 외에도 다양한 방법이 존재한다. 일례로 Peng의 연구에서는 바이너리 기반의 프로토콜 취약성 분석에서 임계치 값(Boundary Value)을 이용한 테스트 케이스를 우선 생성한 바 있다[11]. 그리고 Shahbazi는 테스트 케이스간의 거리가 클수록 더 높은 자유도를 가지는 테스트 케이스 셋이 되고, 결과적으로 높은 코드 커버리지를 확인할 수 있음을 보였다[4]. 이 연구를 통해 자유도가 높은 테스트 슈트가 더욱 효율적이라는 것이 입증되었으나, 스트링 기반의 프로토콜에만 적용 가능하다는 한계점을 가지고 있다. Becker의 연구와 유형욱의 연구는 white-box 테스트의 장점에 Symbolic Execution을 추가한 기법으로 대상 소프트웨어의 분기와 그 분기를 유발하는 값을 탐색하고 이를 토대로 최적화된 테스트 케이스를 생성하여 높은 코드 커버리지를 가지는 장점이 있다[8][9]. 하지만 이 기법은 테스트 진행에 긴 시간이 소모되어 다양한 사물들에 적용하기에는 부적절한 방법으로 사료된다.

Shapiro의 연구는 프로토콜을 알지 못하는 환경에서 가장 최고의 선택이 될 수 있다[3]. 실제 사물인터넷 환경에서 동작할 경우 가장 강력한 효과를 보일 수 있을 것으로 사료되지만, 해당 연구에서는 프로토콜의 필드를 잘 학습 하지 못하여 실제로 활용하기에는 아직 추가 연구들이 필요한 실정이다. 향후 프로토콜 리버싱 기법의 정교함이 증가할수록 더욱 의미 있는 테스트 결과를 도출 할 수 있을 것으로 예상되지만, 대상 소프트웨어나 프로토콜 표준을 기반으로 테스트 케이스를 생성하는 것이 더 뛰어난 테스트 케이스를 생성 할 수 있기 때문에 활용도는 한정

적이다. 사물인터넷 환경의 다양한 프로토콜들은 대부분의 표준이 공개된 상황이기 때문에 표준 기반의 테스트 케이스 생성이 더욱 효과적이다.

퍼징 테스트 기법에 대한 연구들과 함께 최근에 블랙박스 테스트의 단점을 극복하려는 연구들도 진행되고 있다. 이재서의 연구는 테스트 슈트에서 불필요한 테스트 케이스를 제거하는 방법에 대해 논의하며, 코드 커버리지와 테스트 케이스의 길이가 비례한다는 것을 입증하였다[5]. 이 연구 결과를 블랙박스 기반의 퍼징 테스트에 적용할 경우 더욱 효율적인 테스트 진행이 될 것으로 기대된다. 하지만, 해당 연구의 주 대상은 파일 퍼징으로 프로토콜 퍼징 분야에서는 일반적으로 적용되기 어려울 것으로 사료된다.

따라서 기존 연구들이 가지고 있는 한계점과 블랙박스 기반의 테스트에서도 코드 커버리지를 늘리기 위한 연구들에 착안하여 본 논문에서는 무작위성 기반의 테스트 케이스 평가방법과 이를 활용한 사물인터넷 환경에서 취약성 분석을 위한 블랙박스 기반의 퍼징테스트를 방법을 제안한다. 다음 3장에서는 제안하는 퍼징테스트 평가 기법에 대해 논의한다.

III. 테스트 케이스 평가 방법 분석

프로토콜 취약성 분석을 위한 테스트 케이스 평가의 가장 중요한 점은 자유도이다. 자유도가 높은 테스트 케이스 들이 더욱 좋은 테스트 결과를 이끌어 낸다는 Hemmati et al의 연구는 다양한 연구들의 바탕이 되고 있다[13]. 이 연구를 바탕으로 자유도를 기반으로 테스트 케이스를 평가하고, 테스트를 수행하는 다수 연구들이 수행되었다[14][15]. 그러나 위 연구들은 다수의 테스트 케이스 평가를 수행에 오랜 시간이 걸린다는 단점이 있기 때문에 이를 해소하고자 새로운 프로토콜 테스트 케이스 평가 방법을 제안한다.

테스트 케이스 평가 방법에 대한 논의에 앞서 테스트를 진행할 프로토콜이 무엇인지 파악하는 것이 중요하다. 프로토콜은 상태(State)와 형태(Format)으로 구성되어 있다. 프로토콜 상태는 취약성 분석에 아주 중요한 요소로 동일한 입력 값을 대상에게 주입 한다고 하여도 프로그램의 상태에 따라 다른 반응을 보이게 된다. 따라서 상태는 소프트웨어 테스트의 중요한 요소이지만, 프로토콜의 상태가 너무 다양하기 때문에 테스트 케이스 평가 요소에 반영하기는 어렵다. 따라서 본 논문에서는 프로토콜의 형태에 초점을 두고 테스트 케이스 평가를 진행한다.

프로토콜 형태를 보면 하나의 프로토콜은 여러 필드들의 집합으로 구성되어 있고, 각 필드들은 표준에서 정해진 길이의 비트열로 구성되어 있다. 만약 프로토콜이 하나의 필드로만 구성되어 있다고 가정 할 경우 퍼징 테스트를 위해 필드에 어떤 값을 넣는지 판단하는 것이 프로토콜 취약성 분석의 핵심일 것이다. 이를 여러 개의 필드가 존재하는 프로토콜로 다시 확장시켜 생각하면 성공적인 프로토콜 취약성 분석을 위해서는 두 가지의 문제가 존재함을 파악 할 수 있다. 첫 번째는 취약성이 존재할 것으로 판단되는 필드를 선택하는 문제이고, 두 번째는 선택한 필드에 어떤 값을 넣을지 판단하는 문제이다.

첫 번째 문제의 명확한 해답은 존재하지 않는다. 따라서 취약성 분석 테스트는 경험적으로 취약성이 내포되었을 것으로 예상되는 필드를 선택하거나, 모든 필드를 테스트 하는 방법을 선택해야 한다.

반면 두 번째 문제에 대한 연구는 이미 다수 존재한다. A. Arcuri의 연구에 따르면 1차원의 공간에서 오류 탐지를 위해서는 전체 공간에 넓게 펴진 값을 선택하는 것이 가장 확률이 높다는 것을 증명하였다[16]. 이를 확장 시킨 연구들이 앞서 살펴본 거리를 이용한 연구들로 다차원 공간의 거리를 측정하고 이를 바탕으로 테스트 케이스 평가를 수행하였다[4][14]. 하지만 이 기법은 다수의 테스트 케이스를 평가하기 위해서는 너무 많은 연산을 필요로 한다는 문제를 가지고 있어 본 논문에서 새로운 평가 방법을 제안한다.

IV. 무작위성 기반의 테스트 케이스 평가 방법

본 장에서는 앞서 분석한 내용을 토대로 무작위성 기반의 테스트 케이스 평가 방법을 제안한다. 무작위성은 A. Arcuri의 연구에 따라 전체 공간에 넓게 펼쳐진 정도를 나타내는 것으로 본 논문에서 제안하는 기법에서는 분산과 엔트로피를 통해 계산된다. 제안하는 테스트 케이스 평가 방법은 아래와 같다.

4.1 무작위성 계산 : 분산

다시 A. Arcuri의 연구와 같이 하나의 필드를 가지는 프로토콜을 살펴보자. 예를 들어 1Byte 크기의 필드 단 하나만을 가지는 프로토콜이 있다면 이

프로토콜 패킷은 1차원 공간에 있고, 해당 차원은 0x00~0xFF 사이의 값으로 구성된다. 이때 프로토콜 필드에 0x10~0x20 이라는 특정 범위의 값들을 필드에 넣은 테스트 케이스를 만들어 테스트를 수행한다고 가정해 보자. 이 경우 분기(branch)를 유발하는 값이 0xF0인 경우 위 테스트의 모든 테스트 케이스는 해당 분기를 탐색하지 못한다. 해당 분기에 취약성이 존재했을 경우 위 테스트 케이스들로는 파악이 불가능한 것이다.

프로그램은 다양한 분기들로 구성되는데 취약성은 프로그램 동작 중 특정 분기에서 확인되는데 구현상의 취약성이 확인된다. 유발된다. 취약성 분석의 효율성을 증가시키기 위해 소프트웨어 로직 상 분기를 생성할 수 있는 5가지의 관계 연산자들에 주목할 필요가 있다. 따라서 분기를 생성 할 수 있는 총 5가지의 관계연산자 들 중 '같다'를 나타내는 '==' 연산자를 제외한 나머지 관계연산자들로 이루어진 분기는 값의 분포가 커질수록 발견할 확률이 증가하게 된다. 아래 Fig.1.은 간단한 예시로, 특정 함수에서 취약성을 유발하는 분기로 빠지는 테스트 케이스 확보에는 단일 차원일 경우 값들의 분포가 큰 것이 유리하다는 것을 직관적으로 확인 할 수 있다.

Fig.1.의 exploit을 유발하는 분기를 빠르게 찾는 방법이 곧 취약성을 빠르게 발견하는 것으로 연결된다. Chen의 논문에 따르면 테스트 데이터 선택 시 이전 테스트 데이터와 가까이 있는 값 보다 멀리 떨어진 값을 택하는 것이 유의미한 결과를 확인 할 수 있다고 파악되었다[10]. 따라서 무작위성을 대표하는 값으로 분산을 활용하고자 한다.

아래와 같은 수식을 활용 할 경우 분산은 새로운 테스트 케이스가 추가될 경우 소수의 연산만을 통해 새로운 분산을 구할 수 있다. 새롭게 테스트 케이스에 들어갈 값과 기존 분산 및 평균값 그리고 총 시행 횟수를 이용하여 새로운 분산이 계산된다. 기존 분산을 V, 평균을 m이라 할 때, 새로운 값 X_{i+1}을 추가한 새로운 분산 V'과 평균 m'은 다음과 같이 구할 수 있다.

$$V' = \sum_{i=1}^{n+1} \frac{X_i^2}{n+1} - (\sum_{i=1}^{n} \frac{X_i}{n+1})^2$$

$$= \sum_{i=1}^{n} \frac{X_i^2}{n+1} + \frac{X_{i+1}}{n+1} - (\sum_{i=1}^{n} \frac{X_i}{n+1} + \frac{X_{i+1}}{n+1})^2$$

$$= \frac{n \times (V + m^2) + X_{i+1}^2}{n+1} - \frac{n \times m + X_{i+1}}{n+1}$$

$$m' = \frac{m \times n + X_{i+1}}{n+1}$$

4.2 무작위성 계산 : 엔트로피 점수

위와 같이 일반적으로 값의 산포정도를 나타내는 분산을 이용할 경우 빠르게 분기를 확인 할 수 있을 것으로 기대된다. 이를 바탕으로 테스트 케이스의 평가를 수행하는 것도 가능하지만, 분산은 평균을 기준으로 값이 떨어진 정도를 측정하기 때문에 테스트 케이스 들이 얼마나 퍼져 있는지를 정확히 나타내지는 않는다. 일례로 아래 Fig.2.의 두 개의 테스트 셋을 비교하는 경우를 보자.

두 테스트 셋의 분포를 계산할 경우 B가 더 높은 값을 가진다. 그리고 테스트 케이스간의 길이 값 비교를 수행할 경우 A와 B가 동일하게 나온다. 하지만 직관적으로도 테스트 셋 B보다 테스트 셋 A의 값들이 더 퍼져있는 것이 확인된다. 따라서 이 오류를 해결 할 수 있는 다른 방법이 필요하다.

이 오류는 특정 범위에 값이 머물러 있는 경우 발생한다. 따라서 특정 범위에 값이 머물러 있을 때 현저하게 수치가 낮아지는 엔트로피를 이용하여 이 오류를 바로잡고자 한다. 특정 값의 범위에 있는 테스트 케이스들은 유사한 비트열을 가지고 있다. 이에

```
int foo ()
{
  if(flag < threshold){
    normal operation;
  }
  else{
    exploit;
  }
}
```

Fig. 1. Example of program branch

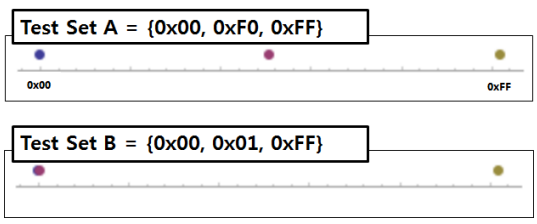

Fig. 2. Example of Protocol Test Set

착안하여 난수 생성기의 무작위성 검증에 사용되는 ApEn Test의 일부를 활용하여 엔트로피를 계산한다[6]. 주어진 대상 비트열을 일정한 길이의 작은 비트열로 나누고, 이 작은 비트열들의 패턴 반복이 얼마나 이루어져 있는지를 토대로 엔트로피를 계산한다. 아래는 본 논문에서 사용하고 있는 엔트로피 계산 방법이다.

우선 테스트 진행에 앞서 반복 패턴을 비교할 작은 비트열의 크기 m을 선택한다. 작은 비트열 크기 m은 결과에 큰 영향을 준다. 너무 작은 값을 선택할 경우 엔트로피 점수가 너무 높게 계산되어 의미가 없어지는 현상이 발생하고, 반대로 테스트 셋의 크기에 비해 너무 크게 선택할 경우 너무 적은 값이 도출되어 값의 의미가 떨어지는 문제가 있다. 따라서 테스트 셋의 크기에 따라 비교할 작은 비트열의 크기를 적절히 정해야한다.

대상 비트열을 T, 반복 패턴을 비교할 작은 비트열의 크기를 m, 작은 비트열이 가질 수 있는 패턴을 모은 집합 X_m이라 한다.

$$X_m = \{X_{m,1}, X_{m,2}, \ldots, X_{m,n}\},\ n = 2^m$$

집합 X_m은 m-bit의 비트열이 가질 수 있는 모든 경우의 수를 포함하고 있기 때문에 n은 2^m으로 계산된다. 이때 $X_{m,i}$가 대상 비트열 T에 존재할 확률을 $P_{m,i}$라 할 때, $P_{m,i}$는 테스트 셋에 존재하는 테스트 케이스를 이용하여 계산 할 수 있다. 이때 각 확률들은 모두 별개이기 때문에 X_m에 대응하는 엔트로피 H_m은 다음과 같이 계산 할 수 있다.

$$H_m = \sum_{i=1}^{n} -P_{m,i} \times \log_2 P_{m,i}$$

H_m은 모든 확률이 동일한 균등분포(uniform distribution)를 이룰 경우 최댓값을 가지게 되고, 한쪽으로 값이 치우칠수록 작은 값을 가지게 된다. 다시 말해, 모든 비트열이 동일한 확률로 존재 할 경우 H_m은 최댓값을 가지고, 한 값에 집중되어 있는 경우 H_m은 최솟값을 가진다.

테스트 셋의 엔트로피는 위와 동일하게 각 테스트 케이스에서 해당 패턴의 존재확률구하고 이를 모두 합하여 테스트 셋 전체에서 각 패턴들이 존재할 확률을 계산한다. 이후 동일한 계산식을 통해 엔트로피 점수(Entropy Score) $Score_E$를 획득한다.

이 엔트로피 점수 값은 대상 T 비트열을 모두 m-bit로 분해하여, 각 요소들이 얼마나 다양하게 구성되었는지를 표현하는 값으로 앞의 분산이 가진 오류를 바로잡기 위해 사용된다. 이 내용을 바탕으로 앞의 Fig.2.의 예시를 확인하면 Table 1.과 같은 결과를 얻을 수 있다.

Table 1. Evaluation Factors of Fig.2.

Set	Variance	Entropy	Distance
A	16256.33	0.501	170
B	21590.33	0.398	170

Table 1.에서 거리는 대상 테스트 셋이 가지고 있는 테스트 케이스들 간 거리의 평균을 사용하였고, 엔트로피 계산에 유의수준은 3으로 계산하였다. 세 요소를 비교해 보면 분산과 거리는 테스트 셋 두 개의 퍼진 정도를 잘못 측정하고 있지만 엔트로피는 이 문제를 완벽하게 해결 할 수 있음을 확인하였다. 하지만 이때 사용된 엔트로피도 테스트 셋 평가에 단독으로 사용되기는 어렵다. 엔트로피 점수는 단순히 비트열이 얼마나 다양한 패턴을 가지고 있는지를 나타내기 때문에 실제 테스트 케이스간 거리와는 무관하다. 따라서 분산과 엔트로피 점수를 곱하여 무작위성 점수(Randomness Score)를 나타낸다.

$$Score_R = Score_E \times Variance$$

위 무작위성 점수를 이용하여 비교하면, 테스트 셋 A의 점수는 10387.8로 테스트 셋 B의 점수인 4685.102보다 높게 나오는 것이 확인 가능하다. 이 점수 값은 분산에 비례하며, 특정 구간에 테스트 케이스가 다수 분포할 경우 분산이 가지는 오류를 바로잡았기 때문에 기존 보다 뛰어난 평가 요소라 판단된다. 다음 장에서는 본 논문에서 제안한 평가 방법에 대한 검증을 수행한다.

V. 실 험

앞서 Fig.2.에서 수행한 것과 동일하게 1차원 공간에서 다수의 테스트 케이스를 비교하여 본 논문에서 제안한 기법의 유효성을 검사한다. 유효성 검사를 위해 무작위로 선별한 30개의 테스트 케이스로 구성된 테스트 셋 4개를 대상으로 비교를 수행한다. Fig.3.은 검사를 위해 사용한 테스트 셋들을 도식화한 것으로 D, B, C, A순으로 값들이 잘 퍼져 있는 것을 볼 수 있다. 이 테스트 케이스들에 대한 각 평

Fig. 3. Sample Test Set Graphing

가 요소의 값은 아래 Table 2.와 같다. 분산과 엔트로피 하나만 사용할 경우 테스트 케이스 평가 요소로 사용이 불가능하다는 것이 확인된다. 그리고 거리의 경우 테스트 셋 B가 가장 넓게 퍼져 있는 것을 잘 파악하였으나, 테스트 케이스 C와 D 중 D가 더 넓게 퍼져 있음에도 불구하고, 반대로 결과를 도출하고 있다. 제안하는 기법은 넓게 퍼진 순서대로 D, B, C, A를 찾아 테스트 케이스 평가에 뛰어남이 확인되었다.

본고에서 제안한 기법이 실제 테스트에서도 효과가 있음을 입증을 위해 산업사물인터넷 부분에서 널리 사용되는 MODBUS 프로토콜을 대상으로 취약성 분석을 수행하였다. 취약성 분석을 위해 프로토콜 라이브러리는 오픈소스인 libmodbus를 이용하여 테스트를 수행하였다[12]. 실험적인 입증을 위하여 제안한 기법을 바탕으로 아래와 같은 간략한 퍼징 테스트 기법을 설계하여 일반적인 기법과의 비교를 진행하였다.

프로토콜의 다양한 필드 중 어디에서 취약성이 발견될지 모르기 때문에 프로토콜이 가지는 필드 모두를 검사한다고 가정하였다. 기존 기법으로 한 필드별에 지정된 수의 테스트 케이스만큼 테스트를 진행하고 다음 필드로 넘어가는 방식의 일반적인 퍼징 테스트를 수행하였다. 반면 본 논문에서 제안한 기법에 따르면 각 필드별 테스트 횟수보다 무작위성 점수가 더 중요한 의미를 가진다. 따라서 제안 기법의 실험으로는 하나의 테스트 케이스를 입력하여 결과를 확인할 때 마다 각 필드 별로 무작위성 점수를 계산하고, 가장 점수가 낮은 필드를 대상으로 다음 테스트 케이스를 생성하고 테스트를 진행하는 방법으로 퍼징 테스트를 설계하고, 테스트를 수행하였다.

Table 3. Number of Tests for specific branch discovery

Number of Test	Traditional	Proposed
Average	101.4	59.4
Maximum	-	142
Minimum	50	6

테스트 도중 라이브러리에 서버가 비정상적으로 세션 종료를 유발하는 문제가 존재하는 것이 확인되었다. 이 문제는 서버의 특정 함수의 분기에서 클라이언트가 응답이 없다고 오인하고 세션을 종료하는 것으로 분석되었다. 이 분기의 유발을 테스트 케이스 평가 기준으로 기존 기법과 제안 기법을 총 25회 반복 수행하여 비정상 세션 종료를 유발하는 분기를 찾는데 걸리는 테스트 횟수를 분석하였다.

테스트 결과는 다음 Table 3.와 같이 발견되었다. 총 150개의 테스트 케이스를 생성하도록 진행한 25회 실험에서 제안 기법은 25회의 실험 중 모두 해당 분기를 확인하였고, 평균적으로 분기 파악에 사용된 테스트 케이스 수가 기존 기법에 비해 약 60%정도 적기 때문에 제안 기법이 기존 기법에 비해 문제가 되는 경로를 찾는데 효과적이라 판단된다.

Table 2. Comparison of Test Case Evaluation Factors

Evaluation Factor	Variance	Entropy	Distance	Randomness
Test Set A	4702.06	0.893	78.74	4198.94
Test Set B	6467.73	0.841	92.58	5439.36
Test Set C	5123.95	0.86	81.03	4406.60
Test Set D	6059.04	0.83	54.50	5029.00

VI. 결 론

본 논문은 엔트로피와 분산을 이용하여 무작위성을 계산하여 테스트 케이스의 평가를 수행하는 방법을 제안하였다. 기존 기법의 테스트 셋이 가지는 테스트 케이스 양에 비례하여 계산 량이 증가하지만, 본 논문에서는 테스트 셋의 크기와 무관하게 빠른 연산이 가능하다. 그리고 제안 기법이 테스트 케이스 평가에 활용될 수 있음을 실험적으로 입증하였다. 그리고 본 논문의 제안 기법이 유효함을 입증하는 과정에서 여러 필드를 대상으로 퍼징 테스트를 수행할 때, 각 필드를 순차적으로 테스트를 수행하는 것 보다 본 논문에서 제안하는 평가 방법을 바탕으로 가장 점수가 낮은 필드에 테스트 케이스를 추가한 경우 약 2배가량 뛰어난 성능이 확인 되었다.

제안하는 기법의 엔트로피 점수를 계산하는 과정 중 비교하는 용도로 사용되는 작은 비트 스트링의 크기선택을 주관적인 연구자의 판단에 의존하는 문제가 있다. 향후 실험적으로 이 내용을 확인하여 테스트 셋의 크기에 따른 적정 수준의 비트 스트링 크기를 파악하는 연구가 필요할 것으로 사료된다.

취약성 테스트가 필요한 기기 및 프로토콜들은 매 순간 증가하고 있다. 이러한 시점에서 본 논문이 제안한 테스트 케이스 평가 방법은 최소한의 테스트 케이스로 최대의 효과를 볼 수 있도록 도와줄 수 있기 때문에 사물인터넷 보안성 향상에 큰 도움이 될 것으로 사료된다. 향후 본 논문의 연구를 기반으로 블랙박스 기반의 취약성 분석 방법을 개발 할 경우 최소한의 테스트 케이스로 취약성을 탐지할 수 있는 기법을 개발 할 수 있을 것으로 기대된다.

References

[1] IoT Security Alliance of KISA, IoT common security guide for security internalization of ICT convergence products and services, KISA, Sep. 2016.

[2] Tahbildar, Hitesh, and Bichitra Kalita. "Automated software test data Generation: Direction of Research," International Journal of Computer Science and Engineering Survey, vol. 2, no. 1,pp. 99-120, Fep. 2011.

[3] Rebecca Shapiro, Sergey Bratus, Edmond Rogers, and Sean Smith, "Identifying vulnerabilities in SCADA systems via fuzz-testing," International Conference on Critical Infrastructure Protection, pp. 57-72, Mar. 2011.

[4] A. Shahbazi and J. Miller, "Black-Box String Test Case Generation through a Multi-Objective Optimization," IEEE Transactions on Software Engineering, vol. 42, no. 4, pp. 361-378, Apr. 2016.

[5] Lee Jaeseo, Kim Jong-Myong, Kim SuYong, Yun Young-Tae, Kim Yong-Min, and Noh Bong-Nam, "A Length-based File Fuzzing Test Suite Reduction Algorithm for Evaluation of Software Vulnerability," Journal of The Korea Institute of Information Security & Cryptology, 23(2), pp. 231-242, Apr. 2013.

[6] L. Bassham, Andrew R., et al, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Sp 800-22, Apr. 2010.

[7] SungJin Kim, Taeshik Shon. "Field Classification based Novel Fuzzing Case Generation for ICS Protocols," Journal of Supercomputing, pp. 1-17, Feb. 2017.

[8] Becker, Sheila, Humberto Abdelnur, Radu State, and Thomas Engel, "An autonomic testing framework for IPv6 configuration protocols," IFIP International Conference on Autonomous Infrastructure, pp. 65-76, Jun. 2010.

[9] Hyunguk Yoo, Taeshik Shon, "Grammar-based Adaptive Fuzzing: Evaluation on SCADA Modbus Protocol," IEEE SmartGridComm, Nov. 2016.

[10] Tsong yueh Chen, Rei-Ching Kuo,

Robert G. merkel, and T.H. Tse, "Adaptive random testing: The art of test case diversity," Journal of Systems and Software, vol. 83, no. 1, pp. 60-66, Jan. 2010.
〔11〕 Peng, S., Cui, B., Jia, R., Liang, S., and Zhang, Y, "A novel vulnerability detection method for ZigBee MAC layer," International Journal of Grid and Utility Computing, vol. 4, no. 2/3, pp. 134-143, Sep. 2013.
〔12〕 Raimbault, S. "libmodbus". Available from http://libmodbus.org/. (Accessed 25 May 2017).
〔13〕 Hemmati, H., Arcuri, A., and Briand, L., "Reducing the cost of model-based testing through test case diversity," International Conference on Testing Software and Systems, pp.63-78, Nov. 2010.
〔14〕 Shi, Q., Chen, Z., Fang, C., Feng, Y., and Xu, B., "Measuring the diversity of a test set with distance entropy," IEEE Transactions on Reliability, vol. 65, no. 1, pp. 19-27, Mar. 2016.
〔15〕 Hemmati, H., Arcuri, A., and Briand, L., "Achieving scalable model-based testing through test case diversity," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 22, no. 6, Feb. 2013.
〔16〕 Arcuri, A., and Briand, L., "Adaptive random testing: An illusion of effectiveness?," 2011 International Symposium on Software Testing and Analysis(ISSTA), pp.265-275, Jul. 2011.

〈저자소개〉

김 성 진 (SungJin Kim) 학생회원
2014년 2월: 아주대학교 정보 및 컴퓨터공학부 공학사
2014년 3월~현재: 아주대학교 컴퓨터공학과 석박사통합과정
〈관심분야〉 스마트그리드 보안, 디지털 포렌식, 네트워크 보안

손 태 식 (Taeshik Shon) 종신회원
2000년: 아주대학교 정보및컴퓨터공학부 졸업(학사)
2002년: 아주대학교 정보통신전문대학원 졸업(석사)
2005년: 고려대학교 정보보호대학원 졸업(박사)
2004년~2005년: University of Minnesota 방문연구원
2005년~2011년: 삼성전자 통신· DMC 연구소 책임연구원
2017년~2018년: Illinois Institute of Technology 방문교수
2011년~ 현재: 아주대학교 정보통신대학 사이버보안학과 교수
〈관심분야〉 ICS/SCADA, DFIR, Anomaly Detection