情報保護學會論文誌
第 23 卷 第 2 號, 2013. 4

일방향 자료전달 시스템의 레거시 서비스 지원을 위한 응답구조 기반 에이전트 자동 생성*

김 경 호,† 장 엽, 김 희 민, 윤 정 한, 김 우 년‡
ETRI 부설연구소

Reply-Type based Agent Generation of Legacy Service on One-way data transfer system*

Kyoung-Ho Kim,† Yeop Chang, Hee-Min Kim, Jeong-Han Yun, Woo-nyon Kim‡
The Attached Institute of ETRI

요 약

물리적 일방향 자료전달 기술은 망간 분리 기술 중 하나로 외부망에서 내부망으로의 데이터 전송회선 자체를 제거하여 외부망을 통한 침입 가능성을 원천적으로 차단한다. 하지만 이로 인해 물리적 일방향 자료전달 기술을 적용하면 양방향 통신을 기반으로 제작된 레거시 서비스를 사용할 수 없다. 레거시 서비스를 운용하기 위해서는 서비스를 지원하는 별도의 에이전트가 필요하다. 하지만 에이전트 개발은 내부 프로토콜 공개, 추가적인 비용 등의 어려움이 존재한다. 본 논문에서는 이런 문제를 해결하기 위해 현장 제어시스템에서 운용 중인 레거시 서비스를 분석하여 세 가지 형태로 분류하였다. 분류를 바탕으로 세 가지 형태의 서비스를 지원하는 에이전트를 자동 생성할 수 있는 방법과 이를 기반으로 한 에이전트 자동 생성 도구의 설계를 제시한다.

ABSTRACT

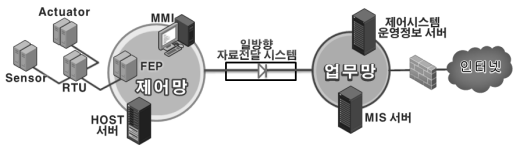
Physical One-way Transfer, one of network Separating Network Technologies, shut off intrusion possibilities by removing data transfer line from external network to internal network. Physical One-way Transfer technology can not support legacy services based duplex transmission. Legacy services operating need agent for extra service with the support. But, Agent development have problems with adding cost and open internal protocols. In this papers, We analyzed legacy services between Control network and OA network in working SCADA systems, and based on the results obtained from the analysis, categorized the legacy services into three forms. We propose an agent generation method of the three service categories for Physical One-Way Transfer System. In addition, we design an automatic generation tool using the proposed method.

Keywords: Physical One-way Transfer System, Legacy Service

I. 서 론

제어시스템은 일반 업무환경에서 사용하고 있는 IT 시스템과는 달리 인터넷에 분리되어 운영되며, 비공개 제어프로토콜을 사용하고, 임베디드 시스템 등을 사용하여 해커나 악성코드에 의해 사이버공격을 받을 가능성이 없다고 인식되어 왔다. 그러나 스턱스넷

접수일(2013년 2월 20일), 수정일(2013년 4월 8일), 게재확정일(2013년 4월 9일)
* 본 연구는 2010년도 지식경제부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (NO. 20101010300091)"
† 주저자, lovekgh@ensec.re.kr
‡ 교신저자, wnkim@ensec.re.kr



〔그림 1〕 제어시스템에 적용된 일방향 자료전달 시스템

(Stuxnet) 악성코드가 2010년 6월 이란의 핵시설 제어시스템에 침투하여 원심분리기의 1/3에 해당하는 1,000여개를 파괴했다는 사실이 알려지고, 2012년도에는 듀크, 플레임, 마호디 등 제어시스템을 대상으로 한 위협이 증가하였다.

최근 제어시스템과 업무망 연계 지점의 보안 위협을 제거하기 위해 일방향 자료전달 시스템의 도입이 추진되고 있다. 제어시스템에서 일방향 자료전달 시스템을 적용하는 위치는 〔그림 1〕과 같다. 일방향 자료전달 시스템〔1〕은 망간 기술의 하나로 업무망에서 제어망으로 데이터를 보낼 수 있는 회선을 제거함으로써, 외부로부터의 침투 경로를 원천적으로 차단한다.

물리적 일방향 자료전달 시스템의 특성상 업무망의 응답 패킷을 제어망에 전송할 수 없다. 이는 양방향 프로토콜에 기반한 레거시 서비스를 지원하는데 많은 어려움을 야기한다. 하지만 기존의 일방향 자료전달 기술〔2,3〕이나 제품〔4,5〕에서 많은 부분 연구 개발이 이루어져 점차 지원이 가능해지는 추세이다. 대표적인 방법이 별도의 서비스 지원 에이전트를 이용하는 것이다.

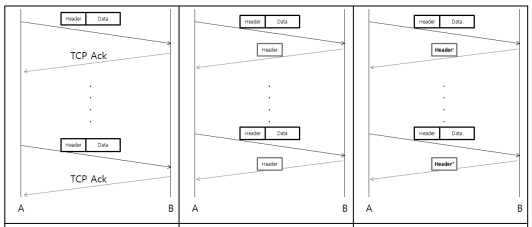
전용 에이전트를 일방향 자료전달 시스템 개발사에 의뢰하는 경우 서비스의 동작 구조와 내부적으로 사용하는 프로토콜을 공개해야하는 부담과 개발에 따른 비용 부담이 존재한다. 이 문제는 일방향 자료전달 시스템의 도입에 큰 걸림돌이 된다.

우리는 현장 제어시스템에서 운용 중인 서비스를 분석하여 3가지의 형태로 분류하였다. 이를 기반으로 본 논문에서는 일방향 자료전달 시스템이 레거시 서비스 지원을 위한 서비스 지원 에이전트를 자동으로 생성할 수 있는 도구를 설계하였다. 본 도구를 이용하면 내부 프로토콜을 외부에 공개하는 부담을 없애고, 에이전트 개발 비용을 줄일 수 있어 일방향 자료전달 시스템 도입을 용이하게 할 것으로 기대된다.

II. 일방향 자료전달 적용 구간 서비스 분석

현장의 제어시스템에는 필요에 의하여 다양한 서비스가 존재한다. 하지만 본 논문에서 대상으로 하는 제어망과 업무망 연계 구간에서 이루어지는 레거시 서비스는 대부분 TCP/IP와 UDP를 이용한다.

이러한 레거시 서비스는 일반적으로 데이터의 일방향성은 확보되어 있지만, 프로토콜의 특성으로 인한 응답 구조가 존재한다. 또 경우에 따라서는 응용 계층에서 별도의 응답 구조가 존재하는 경우도 있다. 본 논문에서는 현장 제어시스템에 운용중인 레거시 서비스를 분석하여 〔그림 2〕와 같이 3가지 형태의 서비스로 분류하였다.



〔그림 2〕 서비스 형태 정의

먼저 TCP/IP와 같은 양방향 프로토콜의 경우 응용 계층이 아닌 프로토콜 내부에 양방향성이 존재한다. 동작 구조는 〔표 1〕의 a. TCP/IP 프로토콜 기반의 형태과 같이 표현할 수 있다. 응용 계층에서 A에서 B로 데이터를 송신만하고 수신하지 않는다. 하지만 프로토콜 내부적인 응답이 존재한다. 이로 인해 일방향 자료전달 시스템을 바로 적용할 수 없다.

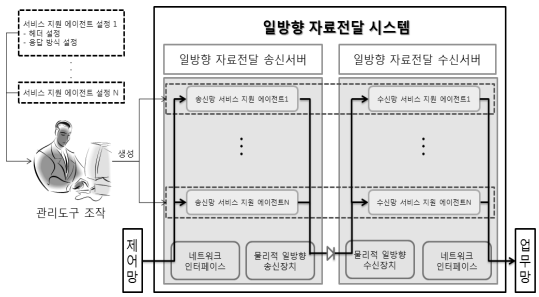
다음은 b. 헤더 응답 기반의 서비스 형태이다. 이 형태의 서비스 역시 TCP/IP 프로토콜을 사용한다. 그리고 여기에 더하여 응용 계층의 응답이 존재한다. b로 정의한 형태에서 사용하는 응용 계층의 응답은 별도의 조작이 없이 수신된 데이터의 특정 부분을 A로 다시 전송하는 형태이다. 현장 제어시스템에서 운용중인 서비스의 경우 다수가 이러한 형태를 취하여 데이터 전송의 신뢰성 확보에 사용한다.

마지막으로 c. 데이터 조작 응답 기반 형태이다. 이 형태의 경우 b. 헤더 응답 기반과 유사한 형태이다. c의 경우는 b의 형태에 추가하여, 수신된 데이터의 일부를 보내는 것이 아닌 수신된 데이터를 기반으로 지정된 로직에 의하여 데이터를 생성하고 이 데이터를 응답으로 하여 A 측으로 전송한다. 이러한 형태의 경우 응답 생성의 모든 로직을 알고, 데이터 생성에 기초가 되는 데이터가 수신한 데이터 내부에 존재해야한

다. 일방향 자료전달 시스템의 구조적인 특성상 외부의 데이터를 이용해야한다면 지원이 불가능하다.

III. 서비스 지원 에이전트 자동 생성 방법

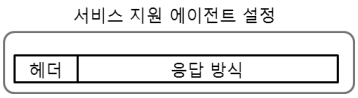
여기서는 Ⅱ절에서 분석한 3가지 형태의 서비스를 지원할 수 있는 서비스 지원 에이전트를 자동으로 생성할 수 있는 방법과 도구의 구조를 제시한다.



〔그림 3〕 서비스 지원 에이전트 자동 생성 개념도

일방향 자료전달 시스템 적용 시 양방향 프로토콜 기반의 서비스를 지원하기 위한 서비스 지원 에이전트의 자동 생성 과정은 〔그림 3〕과 같다.

먼저 Ⅱ절에서 분석된 결과를 바탕으로 해당 서비스에 맞는 헤더 구조 설정, 응답이 존재한다면 응답을 선택 또는 생성할 수 있도록 동작하는 방식을 결정한다. 그리고 자동 생성 도구에 이 정보를 입력하여 서비스 지원 에이전트를 생성하고, 일방향 자료전달 시스템에 탑재한다.

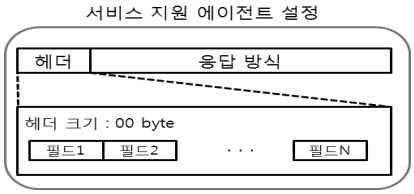


〔그림 4〕 서비스 지원 에이전트 설정 구조도

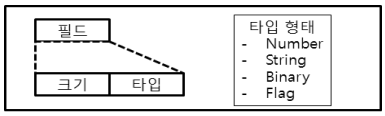
Ⅱ절에서 제시한 3가지 서비스 형태를 지원하기 위해 자동 생성 도구에서 서비스 지원 에이전트를 생성하기 위해 입력해야 할 정보는 〔그림 4〕와 같이 헤더와 응답 방식, 2가지가 있다.

3.1 헤더 설정

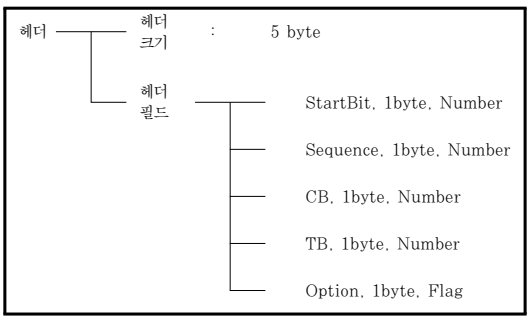
헤더에서는 〔그림 5〕처럼 제어망에서 전송되는 패



〔그림 5〕 서비스 지원 에이전트 설정 - 헤더 구조도



〔그림 6〕 필드 세부 구조도



〔그림 7〕 헤더 설정 예

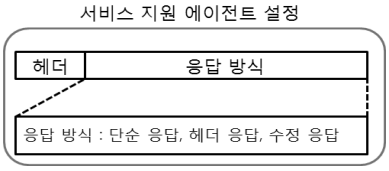
킷 중 헤더의 크기 및 형태를 정의할 수 있다. 또한 응답 기반 및 데이터 조작 응답 기반 형태의 서비스가 제어망으로 전송하는 응답의 형태를 지정할 수 있다. 헤더의 크기와 각 필드의 개수를 정의할 수 있다.

그리고 여기서는 각 필드를 추가 삭제할 수 있고, 필드의 크기와 타입을 설정할 수 있다. 이를 통하여 다양한 형태의 응답을 가지는 여러 서비스를 포용할 수 있도록 설계하였다. 필드의 세부적인 구조는 〔그림 6〕과 같고 〔그림 7〕은 헤더 설정 예이다.

3.2 응답 방식 설정

이 동작을 정의하기 위해서 본 논문에서는 Ⅱ의 분석 결과를 토대로 〔그림 8〕과 같이 단순 응답, 헤더 응답, 수정 응답의 3가지 형태의 응답 방식을 설계하였다. 각 응답 방식의 개념은 〔표 1〕과 같다.

먼저 단순 응답 방식은 TCP/IP 프로토콜을 대상으로 프로토콜 내부적인 응답만 존재하는 서비스를 지원하는 형태이다. 응용 계층에서는 별도의 응답이 존재하지 않기 때문에 응답 방식만 설정하면 서비스 지



〔그림 8〕 서비스 지원 에이전트 설정 - 응답 방식 구조도

〔표 1〕 응답 방식

	단순응답	헤더응답	수정응답
응답 패킷 형태	TCP ack	수신패킷의 헤더	수신패킷의 헤더 수정

원이 가능하다. 헤더 응답 방식은 헤더 응답 기반 서비스를 대상으로 하였다. 응용 계층에서 수신된 데이터 중 지정된 일부를 전송하므로, 이것을 설정할 수 있는 부분을 추가하였다. 마지막으로 수정 응답 방식은 데이터 조작 기반 서비스를 대상으로 하였다. 응용 계층에서 수신된 데이터를 이용하여 지정된 로직에 따라 응답을 생성하고 이를 전송한다.

세부적으로 보면 단순 응답 방식은 특별한 동작이 존재하지 않는다. 따라서 TCP/IP 프로토콜을 이용하여 데이터를 수신하고 전달 할 수 있는 에이전트만 생성하면 된다. 헤더 응답의 경우 수신된 데이터 중 지정된 일부의 데이터를 전송한다. 이를 위하여 헤더 응답은 헤더의 크기, 소스의 위치를 설정한다. 수정 응답 방식은 지정된 로직을 바탕으로 데이터를 생성하여 전송한다.

- 특정 위치의 값을 증가 시켜 전송
- 특정 위치의 값을 감소 시켜서 전송
- 특정 위치 1의 값과 특정 위치 2의 값을 증가하여 전송
- 특정 위치 1의 값과 특정 위치 2의 값을 감소하여 전송

〔그림 9〕 응답 패킷을 위해 지원하는 헤더 수정 방식

본 논문에서는 이러한 수정 응답 방식의 동작 형태를 〔그림 9〕와 같이 4가지로 정의하였다. 따라서 수정 응답에는 크기, 수정할 헤더의 필드, 동작을 설정하여 에이전트를 생성한다.

3.3 서비스 지원 에이전트 생성

3.3.1 TCP/IP 프로토콜 기반

이 형태의 경우는 전송된 데이터에 대한 응용 계층의 응답이 별도로 존재하지 않고, 단순히 프로토콜 내부적인 응답만 존재한다. 자동 생성 도구에서 위의 정보를 바탕으로 송신망용 서비스 지원 에이전트와 수신망용 서비스 지원 에이전트를 각각 설정하여 서비스 지원 에이전트를 생성하게 되면 각각 〔그림 10, 11〕과 같은 코드가 생성된다.

```
while(true)
  //클라이언트 접속 여부 확인
  if(disconnect())
    //클라이언트 접속 종료 전송
    sendFinishOneWay()
    break

  //클라이언트로부터 데이터가 전송되면
  if(receive())
    //수신서버로 전송
    sendOneWay()
return
```

〔그림 10〕 TCP/IP 프로토콜 기반 송신망 서비스 지원 에이전트 의사코드

```
while(true)
  //송신서버로부터 데이터가 수신되면
  if(receiveOneWay())
    //클라이언트 접속 종료가 전송되면
    if(receiveFinishOneWay())
      disconnect()
      return

    //서버 접속
    if(connect())
      //서버로 전송
      send()
return
```

〔그림 11〕 TCP/IP 프로토콜 기반 수신망 서비스 지원 에이전트 의사코드

TCP/IP 프로토콜 기반 형태인 서비스의 경우 단순한 응답 형태를 취한다. 따라서 〔그림 10〕에서 보듯이 접속 여부를 확인하고 일방향 구간으로 수신된 데이터를 전송하는 기능만을 수행한다. 이 형태의 경우 프로토콜 내부적으로 응답이 발생하기 때문에 별도의 응답 로직이 포함되지 않는다.

수신망 서비스 지원 에이전트의 경우 〔그림 11〕에서 보듯이 일방향으로 전송되는 데이터 수신을 대기하고, 수신된 데이터를 바탕으로 업무망과의 연결을 관리하고 원래의 목적지로 데이터를 전송한다.

3.3.2 헤더 응답 기반

헤더 응답 기반 형태의 서비스는 일반적으로 수신된 내용에서 미리 약속된 헤더 부분을 송신 측으로 재전송하는 형태를 취한다.

자동 생성 도구에서 위의 정보를 바탕으로 송신망용 서비스 지원 에이전트와 수신망용 서비스 지원 에이전트를 각각 설정하게 되면 다음과 같은 형태의 서비스 지원 에이전트 코드가 생성된다.

헤더 응답 기반 서비스의 경우 TCP/IP 프로토콜에 기반하고 응용 계층의 응답이 존재한다. 따라서 〔그림 12〕의 코드에 수신된 데이터에서 지정된 부분을 응답으로 전송하는 부분이 추가된다.

```
while(true)
  //클라이언트 접속 여부 확인
  if(disconnect())
    //클라이언트 접속 종료 전송
    sendFinishOneWay()
    break

  //클라이언트로부터 데이터가 전송되면
  if(receive())
    //응답으로 헤더 전송
    ReplyHeader()
    //수신서버로 전송
    sendOneWay()
return
```

〔그림 12〕 헤더 응답 기반 송신망 서비스 지원 에이전트 슈도코드

```
while(true)
  //송신서버로부터 데이터가 수신되면
  if(receiveOneWay())
     //클라이언트 접속 종료가 전송되면
     if(receiveFinishOneWay())
       disconnect()
       return

     //서버 접속
     if(connect())
       //서버로 전송
       send()
       //응답 수신
       receiveHeader()
return
```

〔그림 13〕 헤더 응답 기반 수신망 서비스 지원 에이전트 슈도코드

그리고 〔그림 13〕의 수신망 서비스 지원 에이전트 역시 전송 후 헤더 응답을 대기하는 부분이 추가된다. 이를 통하여 헤더 응답 기반의 서비스를 지원할 수 있다.

3.3.3 데이터 조작 응답 기반

데이터 조작 응답 기반 서비스의 경우 헤더 응답 기반 형태에 더하여 지정된 응답 생성 로직을 이용, 특정 조건에 해당하는 결과물을 응답으로 전송하는 형태를 취한다.

자동 생성 도구에서 위의 정보를 바탕으로 송신망용 서비스 지원 에이전트와 수신망용 서비스 지원 에이전트를 각각 설정하게 되면 〔그림 14, 15〕와 같은 형태의 서비스 지원 에이전트 코드가 생성된다.

```
while(true)
  //클라이언트 접속 여부 확인
  if(disconnect())
    //클라이언트 접속 종료 전송
    sendFinishOneWay()
    break

  //클라이언트로부터 데이터가 전송되면
  if(receive())
    //수신된 데이터 분석
    if(CheckAction())
      //헤더 생성
      MakeReplyHeader()
    //응답으로 헤더 전송
    ReplyHeader()
    //일방향 수신서버로 전송
    sendOneWay()
return
```

〔그림 14〕 데이터 조작 응답 기반 송신망 서비스 지원 에이전트 슈도코드

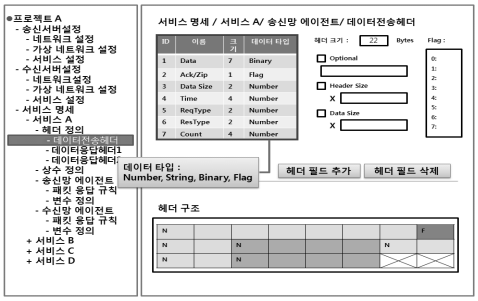
```
while(true)
  //일방향 송신서버로부터 데이터가 수신되면
  if(receiveOneWay())
     //클라이언트 접속 종료가 전송되면
     if(receiveFinishOneWay())
       disconnect()
       return

     //서버 접속
     if(connect())
       //전송할 데이터 분석
       CheckAction()
       //서버로 전송
       send()
       //응답 수신 및 분석
       if(CheckReply(receiveHeader()))
         continue
       else
         return
return
```

〔그림 15〕 헤더 응답 기반 수신망 서비스 지원 에이전트 슈도코드

데이터 조작 기반 서비스의 경우 제어망에서 전송된 데이터를 분석하여 조건에 맞는 응답을 전송하는 형태를 취한다. 이를 위하여 관리자는 서비스의 동작 구조와 응답 생성 로직을 모두 알고 있어야 한다. 자동 생성 도구를 통해 헤더의 구조를 설정하고, 세부적인 동작 방식을 설정하면 정의된 헤더와 동작 방식을 이용하여 레거시 서비스를 지원하게 된다. 물론 전송된 데이터 이외의 데이터를 기준으로 하는 경우는 일방향 자료전달 시스템의 특성상 지원이 불가능하다.

서비스 지원 에이전트 자동 생성 도구 설계

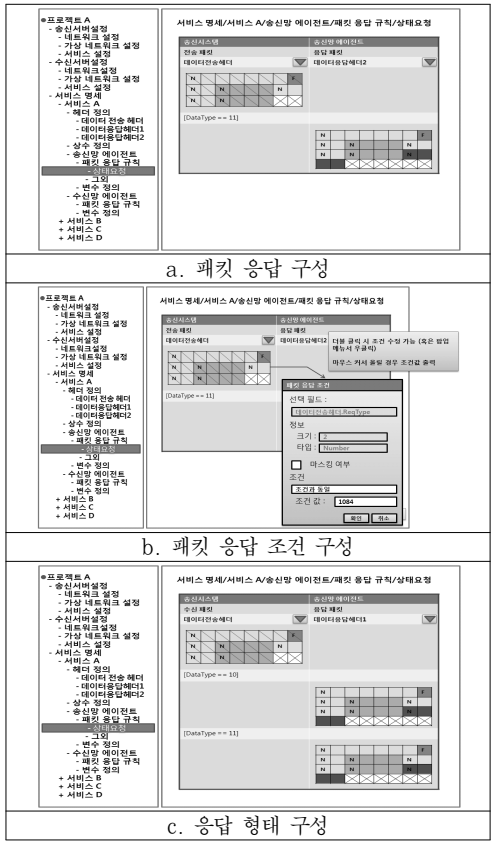
앞에서 제시한 방법을 이용해 서비스 지원 에이전트를 생성하는 자동 생성 도구를 〔그림 16〕과 같이 설계하였다.



〔그림 16〕 생성도구 화면 : 응답 헤더 설정

먼저 헤더 설정을 위해 헤더의 크기와 필드를 설정할 수 있게 하였다. 그리고 위에 언급한대로 필드의 추가 삭제가 가능하도록 하였고, 필드의 크기와 타입을 설정할 수 있도록 하였다.

서비스 설정은 각각 서비스별 사용하는 헤더의 형태를 지정할 수 있도록 하였고, 헤더에 따라 동작을 정의할 수 있도록 하였다. 이를 위해 패킷의 응답 조건을 구성할 수 있는 화면과, 응답 형태 구성, 응답 패킷 규칙을 정의 및 선택할 수 있도록 하였다. 이 도구를 이용하여 〔그림 17〕과 같이 사용자가 쉽게 설정 및 에이전트 생성을 수행할 수 있도록 하였다.



〔그림 17〕 서비스 에이전트 동작 설정

IV. 결 론

본 논문에서는 운용중인 레거시 서비스의 동작 구조와 프로토콜을 분석하여 3가지의 형태로 분류하였다. 그리고 분석된 내용을 바탕으로 서비스 지원 에이전트를 자동 생성할 수 있는 방법을 제안하였고, 자동 생성 도구를 설계하였다. 차후에는 구현 및 성능 평가를 수행하고 더 다양한 형태의 서비스에 대해 지원할 수 있도록 개선할 계획이다.

참고문헌

〔1〕 Pascal Sitbon, Arnaud Tarrago and Pierre Nguyen, "Enabling Secure Information Exchange from a Less Secure Zone to a Control System Zone in a Critical Infrastructure," Proceed-ings of the SCADA Security Scientific Symposium, Digital Bond Press, pp.10, 2003.

〔2〕 Malcolm W.Stevens, "An Implemen-tation of an Optical Data Diode," Def-ence Science And Technology Organisation Caneberra (AUSTRALIA), 1999.

〔3〕 Diego Gonzalez Gomez, "Receive-only UTP cables and Network Taps,"

http://www.infosecwriters.com, 2004.
〔4〕 http://www.waterfall-security.com/category/products/file-transfer-and-replication/
〔5〕 http://www.owlcti.com/process_control/

〈著者紹介〉

사 진

김 경 호 (Kyoung-Ho Kim) 정회원
2010년: 조선대학교 컴퓨터공학과 학사
2012년: 조선대학교 컴퓨터공학과 석사
2011년~현재: ETRI 부설연구소 연구원
〈관심분야〉 패턴인식, 이상징후 감시, 데이터 마이닝, 제어시스템

사 진

장 엽 (Yeop Chang) 정회원
2005년: 고려대학교 컴퓨터공학과 학사
2007년: 고려대학교 컴퓨터공학과 석사
2007년~2010년: LS산전 안양연구소 연구원
2010년~현재: ETRI 부설연구소 연구원
〈관심분야〉 소프트웨어 공학, 소프트웨어 보안

사 진

김 희 민 (Hee-Min Kim) 정회원
2004년: 한남대학교 컴퓨터공학과 학사
2006년: 건국대학교 컴퓨터정보통신공학과 석사
2012년: 건국대학교 컴퓨터정보통신공학과 박사
2011년~현재: ETRI 부설연구소 연구원
〈관심분야〉 네트워크 보안, 네트워크 패킷 포렌식, 차세대 인터넷, IPv6

사 진

윤정한 (Jeong-Han Yun) 정회원
2001년: KAIST 전산학과 학사
2003년: KAIST 전산학과 석사
2011년: KAIST 전산학과 박사
2011년~현재: ETRI 부설연구소 연구원
〈관심분야〉 프로그램 분석, 제어시스템 네트워크 침입 탐지

사 진

김 우 년 (Woo-Nyon Kim) 정회원
1996년: 안동대학교 컴퓨터공학과 학사
1998년: 경북대학교 컴퓨터과학과 석사
2000년: 경북대학교 컴퓨터과학과 박사수료
2000년~2003년: ㈜니츠 선임연구원
2003년~현재: ETRI 부설연구소 선임연구원/팀장
〈관심분야〉 정보보호, 제어시스템 보안