JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 20, No. 4, pp. 835-842, Apr. 2019

프로그램의 정보 보안을 위한 데이터 접근 제어 모델의 설계 및 구현

임 현 일
경남대학교 컴퓨터공학부

Design and Implementation of Secure Data Access Control Model for Improving Information Security of Programs

Hyun-il Lim
Department of Computer Engineering, Kyungnam University, Gyeongsangnam-do 51767, Korea

[요 약]

정보화 시대에서 방대한 정보 및 데이터들을 효과적으로 관리하고 활용하는 것이 필요하며, 데이터 관리 기술은 다양한 응용 분야에서 중요한 부분을 차지하고 있다. 데이터 처리에서 중요한 보안 정보는 외부로 부터의 침임 및 유출로부터 효과적으로 관리할 수 있어야 하며 이런 정보를 보호할 수 있는 기술은 안전한 시스템 운용을 위해서 필수적이다. 접근 제어 모델은 데이터에 허가되지 않은 접근을 차단하고 중요 데이터를 안전하게 보호하기 위한 모델로 활용되고 있다. 본 논문에서는 프로그램에서 사용되는 데이터에 대해서 외부 유출 또는 허가되지 않은 접근으로부터 보호할 수 있는 데이터의 보안 등급을 정의하고, 외부의 접근으로부터 차단할 수 있는 데이터 보안 언어 및 접근 제어 모델을 설계한다. 본 논문에서 제안한 모델은 정의된 보안 데이터를 효과적으로 보호할 수 있음을 실험 결과를 통해 확인한다.

[Abstract]

In the information system, effective management and use of vast amounts of information data play important roles in various applications. To effectively manage information, secure information must be prevented from external intrusion or information leak. So, the technology for protecting secure information against unauthorized access is essential for a safe information system. Access control models can be used to secure data from unauthorized access and to protect against illegal access. This paper presents a data secure language and access control model that can define the security levels of data in a program and protect the data against unauthorized access. The proposed access control model can effectively manage data defined as secure information, and can improve information security of programs. Finally, the accuracy of the proposed model is evaluated through implementation and experiment.

색인어 : 접근 제어 모델, 데이터 보안 프로그램, 데이터 보안, 정보 분석

Key word : Access Control Model, Data Secure Program, Data Security, Information analysis

http://dx.doi.org/10.9728/dcs.2019.20.4.835

Received 25 February 2019 ; **Revised** 13 April 2019
Accepted 26 April 2019

***Corresponding Author; Hyun-il Lim**

Tel: +82-55-249-2650
E-mail: hilim@kyungnam.ac.kr

Ⅰ. 서 론

오늘날 정보 통신 기술은 급속한 발전과 더불어 방대한 데이터를 이용해 다양한 분야에서 활용되고 있다. 네트워크를 기반으로 하는 통신과 4차 산업혁명을 이끌고 있는 인공지능 및 데이터 산업은 정보 통신을 통한 정보화 사회의 무한한 발전 가능성을 보여 준다. 이와 더불어 정보화 사회에서 생산되고 운용되는 방대한 양의 정보들은 빅데이터, 인공 지능과 같은 핵심 분야에서 중요한 역할을 하고 있다.

다양한 데이터를 효율적이고 안전하게 활용할 수 있다면 여러 응용 분야에서 활용할 수 있을 뿐만 아니라 안전한 시스템을 설계하는데 도움이 될 것이다. 반면, 데이터에 따라 사용자의 허가된 정상적인 접근 및 활용 범위를 벗어나는 경우 보안 위험에 노출될 수 있다. 특히, 외부에 노출되거나 접근되는 경우 위험이 발생할 수 있는 민감한 보안 정보의 경우 안전 장치가 필요하다. 따라서, 중요한 데이터를 외부의 허가되지 않은 접근으로부터 안전하게 보호하고 불법적인 접근을 차단하는 보안 모델은 중요한 데이터를 안전하게 보호하기 위한 효과적인 방법으로 고려될 수 있다[1, 2, 3, 4, 5, 6].

본 논문에서는 프로그램에서 사용되는 데이터에 대해서 외부 유출 또는 허가되지 않은 접근으로부터 보호되어야 하는 중요 데이터를 설정하고, 외부의 접근으로부터 차단할 수 있는 데이터 접근 제어 모델을 설계한다. 또한 본 논문에서 제안한 접근 제어 모델은 데이터 보안 언어를 통해 구현하고 실행 결과를 분석한다.

본 논문은 다음과 같이 구성된다. 2장에서 프로그램의 데이터 접근 제어를 통한 보안 향상 방법 및 본 논문의 접근 방법을 소개하고, 이를 위해 필요한 기존의 연구 사례들을 소개한다. 3장에서는 데이터 보안 강화를 위한 데이터 보안 언어를 설계하고 데이터 접근 제어 모델을 통해 데이터 보안 강화 방법을 제안한다. 4장에서는 3장에서 설계한 보안 강화 모델을 프로그램으로 구현하고 데이터 보안성을 위한 프로그램을 통해 본 논문의 데이터 접근 제어 모델의 활용 가능성을 검증한다. 마지막으로 5장에서 본 논문의 결론을 맺는다.

Ⅱ. 데이터 접근 제어를 통한 보안

프로그램에서 사용되는 데이터는 일반적으로 변수 또는 메모리 통해서 전달되고 연산에 사용된다. 정보화 시스템에서 이런 데이터들은 서비스에 필요한 정보를 가지고 있으며 임의로 사용되거나 외부로 유출되지 않도록 안전하게 유지해야하는 데이터의 종류도 증가하고 있다. 예를 들어 시스템 사용자의 개인 정보를 관리하는 과정에서 중요한 정보를 포함하는 고유 번호 등 민감한 정보가 불특정 다수에게 유출되거나 외부의 접근이 이루어진다면 이로 인한 보안 사고가 발생할 수 있다.

Fig 1은 프로그램에서 사용되는 데이터가 외부의 알려지지 않은 대상에 의해서 접근 되거나 유출되는 상황을 보여주고 있다. 이런 경우 프로그램 내의 중요한 정보가 외부 요인에 의해서 변질되거나 유출되는 등의 보안 사고가 발생하게 된다.

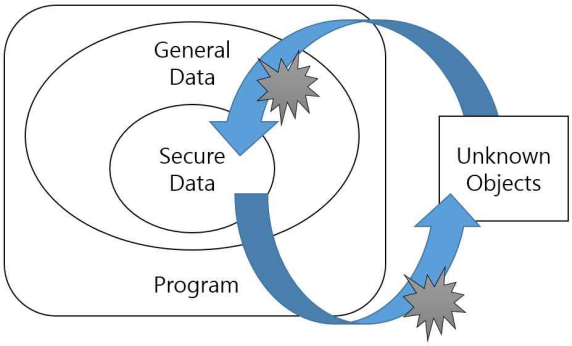


그림 1. 보안 데이터의 위험한 접근 사례
Fig. 1. A case of dangerous access of secure data

이러한 보안 문제에 대응하기 위해서 프로그램에서 사용되는 데이터들을 관리하고 안전하게 보호할 수 있는 보안 모델이 필요하다. 접근 제어 모델은 컴퓨터 시스템에서 다루는 개체에 대해 접근할 수 있는 연산을 보안 규칙에 따라 제어하는 모델을 의미하며, 사이버 보안 강화 [1], 접근 보안 모델의 개선 방안 [2, 3, 6], IoT 보안 [4, 5] 등 다양한 분야에서 연구가 활발히 진행되고 있다. 본 논문에서는 프로그램에서 사용되는 데이터의 중요도에 따라 중요한 데이터는 외부의 위험 요소를 가진 데이터의 접근으로부터 안전하게 관리할 수 있는 접근 제어 모델을 설계한다.

프로그램의 데이터는 프로그램 내에 고정되어 있기 보다는 다양한 연산을 통해서 이동하면서 변화할 수 있다. 따라서 프로그램의 데이터가 어떻게 사용되고 이동하는지 분석할 수 있는 방법이 필요하다. 자료 흐름 분석 [7, 11]은 컴퓨터 프로그램의 동작 지점에서 변수들이 가질 수 있는 값들을 분석하는 기술이다. 이 분석 방법은 주어진 데이터가 어떻게 이동하고 어떻게 이용되는지를 분석을 통해서 파악할 수 있으며, 데이터의 이동 경로 관리 및 추적을 위한 응용 기술로 사용될 수 있다.

동적 테인트 분석 (dynamic taint analysis) [8, 9, 10]은 프로그램의 실행 시간에서 정보의 흐름을 추적하는 방법으로 외부의 부정 입력(malicious input)의 이동 흐름을 분석하고, 이 입력으로부터 발생하는 데이터의 접근이나 변경 가능성을 분석하는 방법이다. 이 방법은 외부의 입력 데이터 흐름을 추적하고 발생 가능한 악성 동작을 파악함으로써 프로그램 실행의 보안을 높일 수 있다.

본 논문에서는 프로그램 내의 중요 데이터의 이동 흐름을 파악하고 알려지지 않은 위험 데이터의 접근을 분석함으로써 위험한 데이터의 접근 및 정보 유출로부터 프로그램을 보호할 수 있는 보안 데이터 접근 제어 모델을 설계한다. 본 논문에서 제안하는 데이터 접근 제어 모델은 프로그램 내의 중요 데이터에 안전한 접근을 보장하는 메커니즘을 제공함으로써 프로그램의 데이터 안전성을 확보하고자 한다.

Ⅲ. 보안 데이터 접근 제어 모델의 설계

3-1 데이터 보안 언어 설계

데이터 접근 제어 모델을 통해 데이터 보안 언어를 설계하기 위해서 범용 프로그래밍 언어에서 따르고 있는 절차적 프로그래밍 [11, 12] 구문을 기반으로 설계하였다. 데이터 보안 언어는 절차적 프로그램에서 사용하는 필수 구문인 순차 구문, 반복 구문, 조건 분기 구문 등을 포함하는 While 언어 [12, 13]에 기반을 두고 있다. While 언어는 절차적 프로그램의 실행에 필요한 대입문, 순차구문, 조건분기구문, 반복구문 등을 포함하는 범용적인 절차적 프로그래밍 언어이다.

Fig 2는 본 논문에서 설계한 데이터 보안 언어의 기본 설계를 보여주고 있다. 전체 보안 프로그램은 한 개 이상의 구문(statement)으로 구성되고, 구문은 연산을 위한 다양한 수식(exp)을 이용해 표현된다. 프로그램에서 사용되는 수식 exp는 상수 또는 논리값을 가질 수 있고, 수식의 연산을 수행하는 단항 연산 ⊗ 및 이항 연산 ⊖을 포함하고 있다. 수식은 다양한 연산자를 통해 명령을 수행할 수 있으며, 프로그램의 외부 입력은 input 명령으로 이루어진다. input 명령은 외부의 출처 src로부터 데이터를 입력 받는 연산을 수행한다.

```
program ::= SecureProgram begin  statement  end
statement ::= { statement }
          | variable = exp
          | statement; statement
          | if ( exp ) then  statement
                    else  statement
          | while ( exp ) do   statement
          | output (variable, dst)
          | setSecurityLevel(variable, SecurityLevel)
exp ::= constant | true  | false
     | variable
     | exp ⊗ exp                 // binary op.
     | ⊖ exp                     // unary op.
     | input (data, src)
SecurityLevel ::= Secure | Normal | Unknown
```

그림 2. 데이터 보안 언어의 설계
Fig. 2. Design of data secure language

데이터 보안 언어의 구문은 절차적 프로그램의 동작에 필요한 대입문, 순차 구문, 선택 구문(if), 반복 구문(while) 등을 포함하고 있다. 그리고, 프로그램의 출력은 출력 명령 output을 통해서 이루어지며, 출력하고자 하는 변수와 출력 대상 dst를 매개 변수로 사용한다.

setSecurityLevel 명령은 데이터 접근 제어 모델을 적용하기 위해서 데이터의 보안 등급(security level)을 지정하는 명령이다. 이 명령어는 보안 중요도를 설정하고자 하는 변수와 보안 등급을 매개 변수로 받아서 변수의 보안 등급을 설정한다. 본 접근 제어 모델에서 변수의 보안 등급은 3단계로 설계하였으며 Table 1에서 보안 등급을 설명하고 있다.

표 1. 데이터의 보안 등급 설계
Table 1. The design of security level of data

security level	description
Normal	• security level of general data • Data is initialized to Normal if there is no explicit security level information.
Secure	• security level of important data to be kept securely
Unknown	• security level of risk data to be maintained separately from secure data • Data originated from unknown source is initialized to Unknown.

Normal 등급은 일반적으로 사용되는 변수의 등급을 나타내며 별도의 등급 설정이 없으면 Normal로 초기화한다. 보안 설정이 필요한 중요 데이터는 setSecurityLevel 명령을 통해 Secure 등급으로 설정하거나 변경할 수 있다. 외부의 알려지지 않은 또는 안전하지 않은 출처로 부터 유입되는 위험 데이터는 Unknown 등급으로 초기화하며, setSecurityLevel 명령을 통해 중요 데이터에 접근을 차단할 수 있다.

3-2 접근 제어 모델의 설계

본 논문에서 제안하는 데이터 보안 언어 및 접근 제어 모델은 접근 제어 리스트 (access control list, ACL) [2, 14]와 두 단계의 접근 제어 알고리즘 accessControl 및 dataAccess로 구성된다. Table 2는 본 논문에서 설계한 데이터 보안 언어의 접근 제어 모델의 구성 요소들을 보여주고 있다.

표 2. 접근 제어 모델의 구성 요소
Table 2. The structure of data access control model

name	description
accessControl	It manages to access data for statements properly in a program by using access control list.
dataAccess	It manages the access rules of data of expressions in programs, and supports accessControl by maintaining access control list.
ACL (access control list)	data structure that manages the value and the security levels for access control for variables or data defined in programs

ACL은 데이터 접근 제어를 수행하기 위해서 프로그램에 사용된 데이터의 보안 정보를 관리하는 자료 구조이다. 이 리스트에서는 접근 제어가 필요한 데이터의 보안 등급과 현재 프로그램 상에서 가지는 값을 관리한다. accessControl 알고리즘은 프로그램의 문장을 한 단계씩 분석하는데, ACL에서 관리하고 있는 보안 데이터에 대한 접근이 규칙에 따라 올바르게 이루어지고 있는지를 분석하고 제어하는 역할을 한다. 또한 문장에서 보안 규칙에 어긋난 데이터 접근을 발견하면 이를 차단하고 사용

자에게 알려주는 역할을 한다. dataAccess 알고리즘은 데이터 접근 규칙이 ACL을 통해 유지될 수 있도록 관리하고, 프로그램에서 사용되는 수식에서 데이터의 접근이 있을 때 접근 규칙에 적합한지 확인하는 역할을 한다. 이 과정에서 Unknown 등급의 데이터가 Secure 데이터에 접근이 탐지되면 accessControl에 알리고 접근을 차단할 수 있도록 지원한다.

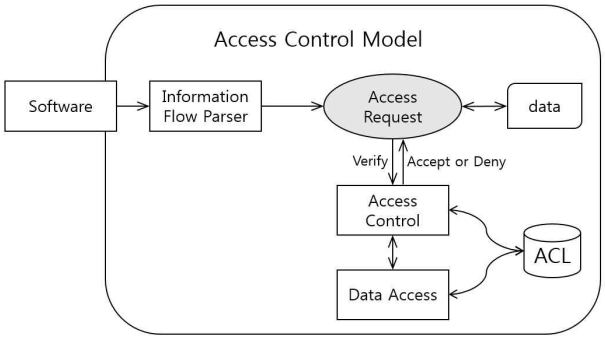


그림 3. 접근 제어 모델의 구조
Fig. 3. The structure of Access control model

Fig 3은 본 논문에서 설계한 접근 제어 모델의 구조를 보여주고 있다. 소프트웨어에 사용되는 보안 정보의 안전한 접근을 보장하기 위해서 파싱된 프로그램에서 정보의 보안 등급을 분석하고 ACL에서 관리되는 보안 정보들은 AccessControl 과 DataAccess에서 분석된 정보의 접근 규칙에 따라 안전성을 확인하고 규칙을 위반하는 경우 접근을 차단할 수 있도록 설계하였다.

3-3 접근 제어 모델의 동작 알고리즘 설계

본 절에서는 본 논문에서 제안하는 데이터 접근 제어 알고리즘에 대해 설명한다. Table 3은 보안 데이터의 접근 제어 규칙을 보여주고 있다. 동등한 보안 등급의 데이터는 서로 접근이 허용(Accept)되고, 서로 다른 보안 등급의 연산이나 출력은 접근 제어 규칙에 따라 보안 등급을 전달하거나 차단한다. Normal 데이터와 Secure 데이터의 연산은 허용하되 결과는 Secure 등급으로 상향하고, Normal 데이터와 Unknown 데이터의 연산은 Unknown 으로 설정함으로서 Secure 데이터의 보안을 유지한다. 또한 Unknown 데이터와 Secure 데이터의 연산이 시도되는 경우에는 경고를 하고 접근을 차단한다.

표 3. Unknown 데이터에 대한 접근 제어 규칙
Table 3. Access control rule for Unknown data

Access rule	Normal	Secure	Unknown
Normal	Accept	-> Secure	-> Unknown
Secure	-> Secure	Accept	*Warning*
Unknown	-> Unknown	*Warning*	Accept

Fig 4와 Fig 5는 Table 2에서 기술한 accessControl과 dataAccess 알고리즘을 보여주고 있다. accessControl 알고리즘은 프로그램의 구문을 분석하며 데이터의 흐름을 파악하고 Secure 등급으로 설정된 데이터가 올바르게 접근되고 있는지 분석한다. 접근 제어 규칙을 위반하는 경우에는 사용자에게 경고를 주고 접근을 차단하는 역할을 한다.

```
accessControl algorithm
stmt = read next statement
while stmt != EOF do
   switch (stmt)
   case variable = exp :
      s_level = dataAccess(exp)
      update the ACL with (variable, s_level)
      stmt = read next statement
   case if ( exp ) then stmt1 else stmt2 :
      val = evaluate the value of conditional exp
      if val == true then
         analyze stmt1 in then part of the if statement
      else
         analyze stmt2 in else part of the if statement
   case while ( exp ) do  stmt1 :
      val = evaluate the value of conditional exp
      if val == true  then
         analyze statement in loop body and continue loop
      else
         analyze next stmt1 of the while loop
   case output (var, dst) :
      s_level = get the security level of var from ACL
      s_dest = dataAccess(dst)
      if security violation from access rule then
         display warning
         block the access to the secure data
      stmt = read next statement
   case setSecurityLevel(var, sl) :
      set the security level of the variable to sl
      update the ACL with (var, sl)
      stmt = read next statement
```

그림 4. 데이터 보안 언어의 접근 제어 알고리즘
Fig. 4. Access control algorithm for data secure language

Fig 4의 접근 제어 알고리즘은 프로그램을 문장 단위로 분석하는데, 데이터의 이동 정보를 파악하고 데이터 접근이 올바른지 판단한다. 분석중인 문장의 종류에 따라 대입문이면 dataAccess 알고리즘을 통해 우변(RHS)의 보안 등급을 알아내고, 대입문의 좌변 (LHS)에 보안 등급을 전파한다. 또한 우변 변수의 보안 등급이 변경되면 ACL을 수정한다. 선택문 if는 조건식 exp의 결과가 참 또는 거짓에 따라 다음 구문을 분석한다. 반복문 while의 경우, 조건식 exp의 결과가 참인 경우는 반복문의 실행을 처리한다.

출력 명령 output은 보안 데이터의 외부 유출 가능성이 있으므로 출력 데이터와 목적지 데이터 dst의 보안 등급을 ACL로부터 확인한다. Secure 등급의 데이터가 Unknown 등급의 목적지 데이터로 출력된다면 데이터 유출 경고를 발생하고, 출력 명령

을 차단한다. 보안 설정 명령 setSecurityLevel의 경우에는 해당 변수의 보안 등급을 새롭게 설정하고, 설정된 정보를 ACL에 반영함으로써 이후에 변수의 접근 제어를 위한 보안 등급을 관리한다.

Fig 5의 dataAccess 알고리즘은 프로그램 내의 수식에 대해 데이터의 접근 규칙에 따라 ACL을 관리하고, 데이터의 접근 제어가 정상적으로 이루어지도록 지원하는 역할을 한다.

```
dataAccess(exp) algorithm
switch (exp)
case  constant or true or false  :
    return Normal as security level of exp
case  var  :
    s_level = get the security level of var from ACL
    return s_level
case  exp1 ⊗ exp2  :
    s_level1 = dataAccess(exp1)
    s_level2 = dataAccess(exp2)
    check access control rule for s_level1 and s_level2
    if security violation then
        display warning
        block the operation with secure data and Unknown
    return s_level1 ⊗ s_level2
case  ⊖ exp1  :
    s_level1 = dataAccess(exp1)
    return s_level1
case  input (data, src)  :
    s_src = dataAccess(src)
    update the ACL with (data, s_src)
    return s_src
```

그림 5. dataAccess 알고리즘
Fig. 5. Algorithm for dataAccess

Fig 5의 알고리즘에서 수식이 상수 이거나 참 또는 거짓의 논리값인 경우 보안 등급을 Normal로 초기화하고 반환한다. 수식이 변수인 경우에는 ACL의 정보를 통해 변수의 보안 등급을 확인하고 이를 반환한다. 수식이 이항 연산인 경우 각 항의 보안 등급을 확인한다. 이 수식이 보안 데이터(Secure)와 의심 데이터(Unknown)의 조합으로 이루어진다면 보안 데이터의 접근 경고를 출력하고 연산을 차단한다. 이와 같은 연산은 개인 번호와 같은 보안 정보를 위변조하거나 외부로 유출하는 등의 방법으로 위험을 야기할 수 있기 때문이다. 단항 연산의 경우에는 포함된 항의 보안 등급으로부터 단항 연산 결과의 보안 등급으로 설정한다. 입력 명령(input)의 경우 데이터의 출처 또는 데이터의 특성으로부터 보안 등급을 확인하고 ACL에 등록한다. ACL은 데이터의 보안 등급을 지속적으로 유지한다.

본 논문의 데이터 접근 제어 모델은 데이터 보안 언어에서 설정된 데이터의 보안 등급을 관리하고, 프로그램에서 사용되는 정보의 이동 및 연산을 분석하면서 의심 데이터로부터 접근을 차단함으로써 보안 데이터를 안전하게 보호한다. 본 논문의 접근 제어 모델은 프로그램에서 데이터의 보안 등급 정의를 통해 프로그램의 중요한 정보를 외부로부터 보호할 수 있다.

Ⅳ. 실험 및 검증

4-1 데이터 보안 프로그램

본 절에서는 본 논문에서 제안한 데이터 보안 언어를 통해 접근 제어 모델을 구현하고 이를 통해서 분석 결과의 유효성을 검증하고자 한다. 본 실험에서는 본 논문에서 제안하고 있는 데이터 보안을 확보할 수 있도록 프로그램에서 데이터의 보안 등급을 설정하고 이를 통해서 보안 데이터가 안전하지 않은 외부의 데이터에 의해서 노출되는 사례를 효과적으로 탐지하고 차단할 수 있는지 평가한다.

```
     SecureProgram begin
 1:    dataA = 100;
 2:    dataB = input(12345, Normal);
 3:    setSecurityLevel(dataB, Secure);
 4:    dataX = input(19216801, Unknown);
 5:    sum = 0;
 6:    setSecurityLevel(sum, Secure);
 7:    sum = sum + dataB;
 8:    i = 0;
 9:    num = 3;
10:    while (i <= num) {
11:      sum = sum + i;
12:      check = sum + dataX;
13:      i = i + 1;
       }
15:    output(sum, dataX);
16:    test = 0;
17:    output(test, dataX);
18:    test = sum + dataX;
19:    output (test, dataX);
20:    if ( dataB > 0 ) then output(dataB, dataX);
                        else output(dataA, dataX);
21:    if ( dataA > 0 ) then output(dataA, dataX);
                        else output(dataB, dataX);
     end
```

그림 6. 데이터 보안 프로그램
Fig. 6. A data secure program

검증 실험을 위해 Fig 2의 데이터 보안 언어를 이용한 데이터 보안 프로그램을 구성하였다. Fig 6은 데이터 보안 언어를 이용한 테스트 프로그램을 보여주고 있다. 본 프로그램의 실행에서 보안 데이터가 여러 연산을 통해서 사용되고 있으며, 실행 과정에서 외부의 Unknown 데이터와 연산이 되거나 유출되는 경우를 포함하고 있다.

테스트 프로그램에서 사용되는 데이터는 Table 4에서 보여주고 있다. dataB와 sum은 라인 3과 라인 6에서 각각 setSecurityLevel 명령을 이용해 보안 등급을 Secure로 설정하였다. 이 변수는 프로그램 내에서 안전하게 보호하고자 하는 보안 데이터에 해당하고, Unknown 데이터로부터 안전하게 보호되어야 한다. 라인 4의 변수 dataX는 외부로 부터 유입된 Unknown 데이터에 해당한다. 따라서 프로그램 내에 사용되는 보안 데이터가 dataX와 함께 연산에서 사용되지 않도록 보장해야 한다. 변수 check는 프로그램의 반복문 라인 12에서 반복적으로 보안 데이터 sum과 dataX 사이에 연산이 이루어진 결과를

대입하는 연산을 수행하고 있으며, 이러한 연산이 탐지되고 제어될 수 있는지 검증한다. 변수 test는 라인 18에서 보안 데이터 sum과 dataX가 함께 연산되고, 이후 연산 결과가 출력 명령 output에서 Unknown 데이터에 의해서 유출되는 사례를 보여준다. 따라서 test 변수의 결과가 외부로 유출되는 경우를 효과적으로 탐지할 수 있어야 한다. 라인 20, 라인 21에서는 if 조건문에서 Normal 데이터 dataA와 Secure 데이터 dataB가 각각 Unknown 데이터 dataX에 의해서 유출되고 있을 때 이를 탐지하고 Secure 데이터의 유출을 정확하게 차단할 수 있는지 검증한다.

표 4. 예제 프로그램에 사용된 데이터 설명
Table 4. The data defined in the example program

name	security level	purpose of the data
dataA	Normal	Normal data
dataB	Secure	Secure data
dataX	Unknown	insecure data from Unknown source
sum	Secure	result value to be kept safely
i	Normal	index var. for loop statements
num	Normal	condition var. for loop statements
check	Secure & Unknown	verify when used in insecure computations in loop statements
test	Secure & Unknown	verify when secure data is exposed to unknown data

Fig 6의 테스트 프로그램에서 보안 데이터가 연산되거나 유출되는 구문은 밑줄로 강조해서 표기하고 있다. 라인 12는 보안 데이터 sum이 Unknown 데이터 dataX와 연산되어 값이 변조되거나 유출되는 경우에 해당한다. 이 문장은 반복문내에서 반복적으로 연산이 수행된다. 라인 15에서는 보안 데이터 sum이 Unknown 데이터 dataX와 함께 출력 명령 output을 통해 외부로 유출되는 경우이다. 라인 18은 보안 데이터와 Unknown 데이터의 연산 결과가 Normal 데이터 test에 대입되고 있다. 따라서, test는 보안 데이터의 값을 포함하고 있으며, 이 과정에서 Table 3의 접근 제어 규칙에 따라 Secure 보안 등급으로 변화된다. 따라서 라인 17과 라인 19는 동일한 output 명령이지만 라인 17에서 test는 Normal 등급을 가지고 데이터 유출에 자유로울 수 있지만 라인 18의 연산에서 test는 Secure 데이터로 변경됨에 따라 라인 19의 output 명령은 유출되어서는 안되는 경우를 보여준다. 라인 20과 21은 if 조건문에서 보안 데이터 dataB가 output 명령에 의해 외부 유출되는 상황을 나타내고 있으며, 라인 21의 경우 조건문의 결과에 따라 실제 데이터 유출 명령은 실행되지 않는 것을 알 수 있다.

4-2 실험 결과

본 논문에서 제안한 접근 제어 모델은 데이터 보안 언어를 통한 구현 및 실험을 하고 정확성을 검증하였다. 본 구현 환경은 Microsoft Windows 7 운영 체제에서 함수형 언어 하스켈(Haskell) [15]을 이용해서 구현하였다. 함수형 언어 하스켈은 문자열을 다루는 고급 기능들을 지원하기 때문에 프로그래밍 언어의 설계 및 분석기 개발 등에서 효과적인 프로그래밍 환경을 제공한다.

```
  [Dangerous Access Output Warning] :

("SECURE DATA [sum] is Accessed by Security Level:
Danger",(VConst 12351,Danger))
("SECURE DATA [test] is Accessed by Security Level:
Danger",(VConst 19229152,Danger))
("SECURE DATA [dataB] is Accessed by Security Level:
Danger",(VConst 12345,Danger))

  [Dangerous Access: Detailed Information] :

1:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTApproach]
        ID: [*]   Value: [12345]   Security: Secure
        ID:  [*]     Value:  [19216801]     Security:
Unknown
2:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTApproach]
        ID: [*]   Value: [12346]   Security: Secure
        ID:  [*]     Value:  [19216801]     Security:
Unknown
3:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTApproach]
        ID: [*]   Value: [12348]   Security: Secure
        ID:  [*]     Value:  [19216801]     Security:
Unknown
4:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTApproach]
        ID: [*]   Value: [12351]   Security: Secure
        ID:  [*]     Value:  [19216801]     Security:
Unknown
5:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTOutput]
        ID: [dataX]    Value: [19216801]   Security:
Unknown
        ID:  [sum]     Value:  [12351]     Security:
Secure
6:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTApproach]
        ID: [*]   Value: [12351]   Security: Secure
        ID:  [*]     Value:  [19216801]     Security:
Unknown
7:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTOutput]
        ID: [dataX]    Value: [19216801]   Security:
Unknown
        ID: [test]    Value: [19229152]    Security:
Danger
8:[Warning] Secure data is used with Dangerous Data.
    Danger Type: [DTOutput]
        ID: [dataX]    Value: [19216801]   Security:
Unknown
        ID: [dataB]     Value:  [12345]     Security:
Secure
```

그림 7. 데이터 보안 프로그램의 접근 제어 실행 결과
Fig. 7. Results of evaluation with the test program

구현된 데이터 보안 언어를 통해 Fig 6에서 제시한 데이터 보안 프로그램을 이용해 접근 제어 모델의 분석 결과를 실험하였다. Fig 7은 구현한 모델을 통한 프로그램의 분석 실행 결과를 보여주고 있으며, 보안 데이터의 위험을 탐지하고 차단할 수 있도록 정보를 제공하고 있다. 첫 번째 Output Warning 분석 결

과는 output 명령을 통해서 Secure 데이터가 Unknown 데이터를 통해서 외부로 유출이 탐지되는 경우에 대한 분석 결과를 경고하고 있다. 이 분석 결과는 Fig 6의 프로그램에서 라인 15, 18, 20에서 사용된 보안 데이터의 output 명령에 대해서 데이터 유출 사례를 정확하게 탐지하였다.

두 번째 Access 분석 결과에서는 Secure 데이터에 대한 Unknown 데이터의 접근 및 연산(분석 타입: DTApproach)과 output 명령을 통한 보안 데이터 유출(분석 타입: DTOutput)을 탐지하고 경고 메시지를 보여준다. 경고 메시지에는 위험 분석 타입, 사용되는 데이터의 현재 값, 그리고 데이터의 보안 등급을 보여주고 있다. 1 ~ 4번 경고 메시지는 반복문 내 라인 12에서 Secure 데이터 sum이 Unknown 데이터 dataX와 함께 연산이 되면서 데이터의 유출 및 위조 가능성을 경고 하고 있다. 총 4회의 반복문이 수행되면서 4회의 명령 수행에 대해서 데이터의 현재 값과 보안 등급 등의 분석 결과를 보여준다. 5, 6, 8번 경고 메시지는 라인 15, 19, 20에서 output 명령에 의해서 보안 데이터의 외부 유출이 시도되는 경우이다. 7번 경고 메시지는 라인 18에서 보안 데이터 sum이 위험 데이터 dataX와 함께 연산이 되는 상황을 탐지하고 있다. 본 명령에 의해서 변수 test는 보안 등급이 Normal에서 Secure로 조정된다. 따라서 라인 17에서 사용된 output 명령과 달리 라인 19에서 사용된 output 명령은 보안 데이터의 유출 사례로 탐지되고 있다.

본 실험 결과는 본 논문에서 설계한 데이터 보안 언어로 작성된 프로그램에서 Secure 데이터로 설정된 dataB, sum이 Unknown 데이터로 사용되는 dataX와 함께 연산이 되거나 output 명령으로 외부 유출되는 사례에 효과적으로 탐지한 결과를 보여준다. 또한, 테스트 프로그램의 라인 18에서 데이터의 연산을 통해 변수 test의 보안 등급은 Normal에서 Secure로 상승하였으며, output 명령에 의해서 유출되는 경우에 정확하게 Secure 데이터에 대해서 탐지함을 확인할 수 있다.

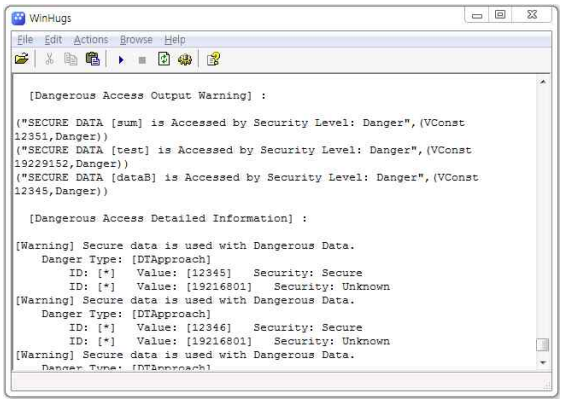


그림 8. 보안 프로그램의 접근 제어 실험 실행 결과
Fig. 8. Execution results of the experiments

Fig 8은 본 논문에서 구현하고 실험한 데이터 접근 제어 모델에 대한 실행 화면을 보여준다. 프로그램 내에서 사용되는 데이터의 안전한 유지 및 관리는 정보 보안의 중요성이 커짐에 따라 다양한 연구를 통해서 확보되어야 하는 기술이다. 본 논문에서는 접근 제어 모델을 통해 프로그램상의 보안 데이터를 안전하기 유지하고 보호할 수 있는 체계를 확보할 수 있는 방법을 제시하고 있다. 제안된 접근 제어 모델은 구현 및 실험을 통해서 보안 데이터의 효과적인 탐지 및 차단이 가능한 것을 확인할 수 있다.

Ⅴ. 결 론

오늘날 정보 통신 기술의 급속한 발전과 더불어 정보화 사회에서 생산하고 응용할 수 있는 방대한 양의 정보들은 빅데이터, 인공 지능과 같은 핵심 분야에서 중요한 역할을 하고 있다. 반면, 데이터 소유자의 정상적인 접근 및 활용 범위를 벗어나는 경우 다양한 보안 위험에 노출될 수 있다. 특히, 민감한 보안 정보가 외부에 유출되는 경우 심각한 피해가 발생한다. 본 논문에서는 중요한 데이터를 외부의 허가되지 않은 접근으로부터 안전하게 보호하고 불법적인 접근을 차단하는 보안 모델을 제안하고 있다.

본 논문에서는 프로그램에서 사용되는 데이터에 대해서 외부 유출 또는 허가되지 않은 접근으로부터 보호되어야 하는 보안 등급을 정의하고, 보안 데이터를 외부 접근으로부터 차단할 수 있는 데이터 보안 언어와 접근 제어 모델을 설계하고 실험을 통해 보안 데이터를 이용한 프로그램에서 위험 데이터의 접근을 효과적으로 차단할 수 있음을 보여준다.

정보화 사회에서 데이터 사용량은 기하급수적으로 증가하고 있으며, 데이터 관리 기술은 필수 기술이 되고 있다. 본 논문에서 제안한 접근 제어 모델은 프로그램에서 데이터를 안전하고 보호하고 외부로의 유출을 차단할 수 있는 기술 분야에 효과적으로 활용될 수 있으며, 본 모델은 데이터 보안 유지를 강화함으로써 안전한 데이터 관리를 필요로 하는 보안 시스템의 설계 및 구현에 활용할 수 있을 것으로 기대된다.

감사의 글

이 연구결과물은 2017학년도 경남대학교 학술진흥연구비 지원에 의한 것임.

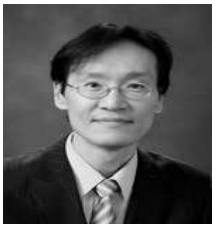
참고문헌

[1] Romuald Thion, Access Control Models, Chapter 37, *Cyber Warfare and Cyber Terrorism*, 2007.

[2] John Barkley, Comparing simple role based access control models and access control lists, *The second ACM workshop on Role-based access control*, pages 127-132, 1997.

[3] Chang N. Zhang, Cungang Yang, Information flow analysis on role-based access control model, *Information Management & Computer Security*, Vol. 10 Issue: 5, pp.225-236, 2002.

[4] Luis Cruz-Piris, Diego Rivera, Ivan Marsa-Maestre, Enrique de la Hoz, and Juan R. Velasco, Access Control Mechanism for IoT Environments Based on Modelling Communication Procedures as Resources, *Sensors*, Vol 18, No. 3, Mar. 2018.
[5] Mehdi Adda, Jabril Abdelaziz, Hamid Mcheick, Rabeb Saad, Toward an Access Control Model for IOTCollab, *Procedia Computer Science*, Volume 52, Pages 428-435, 2015,
[6] Srdjan Marinovic, Robert Craven, Jiefei Ma, Naranker Dulay, Rumpole: A flexible break-glass access control model. *The 16th ACM Symposium on Access Control Models and Technologies*. pages 73-82. 2011.
[7] Uday Khedker, Amitabha Sanyal, and Bageshri Sathe, *Data Flow Analysis: Theory and Practice*, CRC Press, 2009.
[8] James Newsome and Dawn Song. Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software. *The Network and Distributed System Security Symposium*, 2005.
[9] Winnie Cheng, Qin Zhao, Bei Yu, and Scott Hiroshige, TaintTrace: Efficient Flow Tracing with Dynamic Binary Rewriting, *The 11th IEEE Symposium on Computers and Communications*, pp. 749-754, June 2006.
[10] James Clause, Wanchun Li, and Ro Orso, Dytan: A Generic Dynamic Taint Analysis Framework, *The International Symposium on Software Testing and Analysis*, pp. 196-206, 2007.
[11] Arvind Kumar Bansal, *Introduction to Programming Languages*, Chapman and Hall, 2013.
[12] Kenneth Slonneger and Barry L. Kurtz, *Formal Syntax and Semantics of Programming Languages*, Addison Wesley, 1995.
[13] Massimo Merro, The WHILE programming language, http://profs.sci.univr.it/~merro/files/WhileExtra_l.pdf
[14] Messaoud Benantar, *Access Control Systems: Security, Identity Management and Trust Models*, Springer, 2006.
[15] Programming Language Haskell, http://www.haskell.org/



임현일(Hyun-il Lim)

1995년 : KAIST 전산학과 (공학사)
1997년 : KAIST 전산학과 (공학석사)
2009년 : KAIST 전산학과 (공학박사)

2009년～2010년: KAIST 전산학과 연구원
2010년～현 재: 경남대학교 컴퓨터공학부 부교수
※관심분야 : 소프트웨어 분석, 소프트웨어 보안, 인공 지능, 기계 학습, 소프트웨어 공학, 프로그래밍 언어 등