KNOM Review '17-01 Vol.20 No.01
https://doi.org/10.22670/knom.2017.20.1.9

SCADA 시스템의 주기성과 발생순서 정보를 반영한 Structured ACL 모델에 관한 연구

정 우 석◆, 윤 정 한*, 김 신 규**, 구 영 훈***, 이 성 호****, 김 명 섭°

A Study on Structured ACL Model Reflecting Periodicity and Sequence of SCADA System

Woo-Suk Jung◆, Jeong-Han Yun*, Sin-Kyu Kim**, Young-Hoon Goo, Sung-Ho Lee, Myung-Sup Kim°

요 약

SCADA 시스템은 정해진 기기들이 제한된 통신만을 반복적이고 주기적으로 사용하며 동작한다. 이러한 특징 때문에 화이트리스트 기반 보안기법이 많이 사용되고, 실제 현장에 가장 많이 적용되어 있는 것이 화이트리스트 기반 Static ACL을 이용한 접근제한 기법이다. Static ACL은 보안효과가 높은 장점이 있지만 표현력이 너무 단순하여 Dynamic Allocated Port를 사용하는 통신을 표현하는데 한계를 가진다. 또한 제어기기의 통신 특성을 전부 반영하지 못하고, 생성된 Static ACL은 분기 별로 한번 사용되더라도 1년 365일 오픈해야한다는 세 가지의 한계점을 가진다. 우리는 Static ACL에 SCADA 시스템의 기계적이고 반복적인 통신특성을 반영하기 위하여 통신별 주기성 그리고 통신간의 고정된 발생순서 정보를 확장한 Structured ACL을 제안하였다. 제안하는 시스템은 실제 SCADA 네트워크 트래픽에 적용을 통하여 타당성을 증명하였다.

Key Words : Industrial Control System, SCADA, Whitelist, Traffic Locality, Frequent Pattern Mining

ABSTRACT

The SCADA system works by repeatedly and periodically using only limited communication devices. Because of this feature, whitelist based security techniques are widely used, and access restriction method using whitelist based static ACL is most commonly applied in the field. Static ACL has an advantage in security, but their expressiveness is too simple to express communication using dynamic allocated port. In addition, it does not reflect all the communication characteristics of the control device, and the generated static ACL should always be open regardless of the frequency of use. We have proposed a structured ACL that extends the fixed generation sequence information between the communication and communication-specific periodicity to reflect the mechanical and repetitive communication characteristics of the SCADA system in the static ACL. We demonstrate the feasibility of the proposed system in this paper by applying the real SCADA network traffic.

※이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 기초연구사업(No.2015R1D1A3A01018057) 및 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2017-0-00513)
◆ First Author : Korea University Department of Computer and Information Science, hary5832@korea.ac.kr
° Corresponding Author : Korea University Department of Computer and Information Science, tmskim@korea.ac.kr
* National Security Research Institute, dolgam@nsr.re.kr
** National Security Research Institute, skkim@nsr.re.kr
*** Korea University Department of Computer and Information Science, gyh0808@korea.ac.kr
**** Korea University Department of Computer and Information Science, gaek5@korea.ac.kr
논문번호 : KNOM2017-01-02, Received April 20, 2017; Revised May 25, 2017; Accepted July 20, 2017

Ⅰ. 서 론

제어시스템은 특정 산업현장 전체 또는 산업 단지를 전반적으로 감시하고 제어하기 위하여 다양한 기간 시설과 산업에서 사용되고 있는 컴퓨터 기반의 시스템이다. 제어시스템 중 SCADA(Supervisory Control And Data Acquisition) 시스템은 1960년대에 원격지의 시스템을 감시하고 제어하기 위하여 사용되기 시작하였다. SCADA 시스템은 폐쇄망에서 비공개 프로토콜을 사용해 동작하였기 때문에 지금까지 보안과 관련한 위협이 크지 않았다. 하지만 최근 비즈니스 시스템과의 통합으로 인해 네트워크가 확산됨에 따라 폐쇄망과 업무망 그리고 인터넷망 간의 연결이 증가하였으며, 외부 협력업체와의 협업으로 인한 연결성과 다양한 매체의 이용 등으로 인한 보안상 취약점들이 증가하고 있는 추세이다. 화이트 리스트 보안 기법은 안전이 증명된 것만을 허용하는 것으로 악의성이 입증된 것들을 차단하는 블랙리스트(blackllist) 보안 기법과 상반되는 보안 방식이다. 화이트리스트는 보안성은 높지만 편의성을 심각하게 저해할 수 있어 제한적인 영역에서만 사용되었지만, 특히 리소스가 적고 시스템 동작 패턴 또는 네트워크 통신 트래픽이 규칙적인 제어시스템 환경에서 보안을 담보할 수 있는 효율적 방안으로 주목 받고 있다.

SCADA 네트워크는 정해진 기기들이 제한된 통신만을 반복적이고 주기적으로 사용하며 동작한다. 하지만 현재 화이트리스트 보안 기법에서 일반적으로 사용되는 Static ACL은 단순한 5-tuple 정보만을 포함하므로 SCADA 시스템의 기계적이고 반복적인 통신특성을 반영하지 못한다는 한계점을 가진다.

본 논문에서는 SCADA 시스템의 기계적이고 반복적인 통신특성을 반영하기 위해서 Periodicity와 sequence의 두 가지 특징을 반영한 ACL(Access Control List) 간의 순서 연관관계를 추출한다. 추출된 순서 연관관계는 Static ACL에 추가하여 SCADA 시스템의 기계적 통신 특성이 반영된 보다 구체적인 화이트리스트인 Structured ACL를 구성한다.

2장에서는 관련 연구에 대해 서술한다. 3장에서는 기존 Static ACL이 가지는 한계점과 본 논문에서 제안하는 Structured ACL을 구성하는 다양한 ACL들에 대해 정의한다. 4장에서는 제안하는 시스템에 대해 서술하고, 5장에서는 실험과 결과를 6장에서는 결론 및 향후 연구에 대해 차례로 서술한다.

Ⅱ. 관련 연구

Static ACL은 네트워크 스위치, 방화벽 등 다양한 보안제품에서 사용된다. 국제적인 정보보호 기업인 CISCO, Juniper Networks 그리고 McAfee 등은 Static ACL을 자신들이 만든 네트워크 스위치, 방화벽 등에 적용하고 있다. 또한 무선랜 시장에서 화두가 되었던 ArubaNetworks 역시 무선랜 스위치에 Static ACL을 사용하고 있다. 이처럼 Static ACL은 제어시스템 뿐만 아니라 다양한 곳에서 활용되고 있다. 하지만 개별 장비에 적용하기 위해 사람이 Static ACL을 작성하는 것은 매우 어렵기 때문에 Static ACL을 자동으로 생성하는 방법에 대한 다양한 연구가 진행 되고 있다[2][8].

Choi Seungoh et al[2]의 연구에서는 SCADA 시스템의 TCP 핸드쉐이킹이나 공통된 포트 등의 일반적인 특성이 플로우 기반 화이트리스트를 생성하는데 불충분한 점을 문제로 제기하며, 트래픽의 지역성을 기반으로 한 플로우 화이트리스트를 자동으로 생성하는 방법을 제안했다. 해당 연구에서는 플로우의 5-tuple 정보만을 사용하여 Locally Frequently-Used Port와 Degree Centrality를 활용하여 화이트리스트를 자동으로 생성하고 이를 실제 SCADA 시스템 트래픽에 적용하여 검증하였다.

하지만 기존의 Static ACL 자동 생성에 관한 연구를 통해 생성된 Static ACL은 SCADA 네트워크의 특징을 표현하는데 있어 표현력에 한계가 존재한다. 또한 ANY-ANY 같은 규칙이 많이 생성된다는 점과 한 번만 발생한 통신이라도 규칙으로 생성한다는 등의 단점이 존재한다.

본 논문에서는 기존 Static ACL 자동 생성에 관한 연구가 가지는 Static ACL 표현력의 한계와 위에서 언급한 단점을 극복하기 위하여, Static ACL에 순서 연관관계를 추가하여 SCADA 시스템의 특징을 반영한 Structured ACL을 제안한다.

Ⅲ. Static ACL과 그 한계점

본 장에서는 기존 Static ACL 모델에 대하여 정의하고, Static ACL 모델이 가지는 세 가지 한계점에 대하여 서술한다.

1. Static ACL

일반적으로 화이트리스트를 표현하는 Static ACL은 5-tuple 기반의 단순한 네트워크 및 시스템에 대한 접근 권한 리스트이다. 따라서 단순한 5-tuple 정보만을 포함하고 있으며, ACL간의 순서나 조건을 표현 하지 못한다.

2. Static ACL의 한계점

현재의 화이트리스트를 사용한 제어시스템의 보안 기법은 대부분 Static ACL 모델을 사용하고 있다. 하지만 Static ACL 모델을 기반으로 한 화이트리스트는 세 가지 한계점을 가지고 있다.

첫 번째 한계점은 SCADA 시스템에서 사용되는 FTP나 OPC(OLE for Process Control)와 같은 Dynamically Allocated Port를 사용하는 통신의 경우 Static ACL 모델로 표현하는 방법은 ANY-ANY 규칙을 사용하는 것이 유일하다는 것이다. 하지만 ANY-ANY 규칙을 허용하게 되면 해당 IP 간의 모든 연결에 대하여 모든 포트를 항시 오픈해야한다. 이는 기존 해당 IP 간에 생성된 다른 ACL 규칙들이 무의미 해지게 되는 것을 의미한다. 특히 FTP를 사용하는 서버는 해당 서버에 접근하는 모두에게 오픈 될 가능성이 있다.

두 번째 한계점은 제어기기의 통신 특성을 전부 반영하지 못 한다는 점이다. 제어기기는 매일 계속 정해진 작업을 반복적으로 수행해야 하는 기기인데, Static ACL 모델을 사용하면 정해진 작업을 수행한다는 조건만 표현된다.

세 번째 한계점은 빈도에 상관없이 Static ACL 모델로 작성된 모든 규칙은 항시 오픈 된다는 점이다. 예를 들어 SCADA 네트워크 장비 관리 서비스에 대한 관리 단말 지정의 경우 1년에 몇 회 사용하지 않지만 1년 내내 오픈해 주어야 한다는 점이다.

Ⅳ. 제안하는 Structured ACL 모델 및 추출 시스템

본 장에서는 Static ACL이 가지는 세 가지 한계점을 극복하기 위하여 본 논문에서 제안하는 Structured ACL 모델과 이를 구성하는 다양한 ACL들에 대해서 정의한다. 또한 Static ACL 간의 연관관계를 추출하고 이를 Structured ACL로 표현하기 위한 방법에 대해 기술한다.

1. Structured ACL Model

본 논문에서는 위에서 서술한 Static ACL이 가지는 세 가지 한계점과 제어시스템 트래픽의 특징을 표현하는 표현력을 확장한 Structured ACL을 정의하고 사용한다.

Structured ACL은 제어시스템 네트워크 트래픽으로부터 연관관계를 추출하여 정의 된 순서를 가지는 ACL의 집합으로 빈번한 Static ACL들을 추출한 Frequent ACL, Static ACL들의 주기를 추출한 Periodic ACL, Static ACL들 간의 연관관계를 추출한 Sequential ACL 그리고 세 가지 ACL을 통해 추출 되지 않은 Static ACL의 4 가지 ACL을 통해 구성된다. Structured ACL과 Structured ACL을 구성하는 Static ACL, Frequent ACL, Periodic ACL, Sequential ACL은 표 1과 같이 정의된다.

표 1. ACL 정의
Table 1. Definition of ACL

용어	정의
ACL	사용자들이 디렉토리나 파일과 같은 특정 시스템에 접근 할 수 있는 권한을 설정해 놓은 리스트
Static ACL	5-tuple 기반의 단순한 네트워크 및 시스템에 대한 리스트
Frequent ACL	빈번한 Static ACL의 반복 주기를 추출한 리스트
Periodic ACL	Static ACL의 반복 주기를 추출한 리스트
Sequential ACL	Static ACL 사이의 순서나 조건 등의 연관관계를 추출한 리스트
Structured ACL	Static ACL 사이의 순서나 조건 등의 연관관계를 표현한 리스트 {Static ACL , Frequent ACL, Periodic ACL, Static ACL}

Frequent ACL과 Periodic ACL은 모두 ACL의 주기를 추출한 ACL이다. 하지만 실제 트래픽을 분석한 결과 아주 자주 사용하는 ACL에 의해 sequence가 너무 많이 추출되는 문제가 발생하였고, 불필요한 sequence를 제거하고 너무 빈번히 발생하는 ACL들이 Structured ACL로 구성되어 오히려 시스템에 부하를 발생하는 경우를 방지하기 위하여 Frequent ACL과 Periodic ACL을 구분하였다.

그림 1은 Static ACL, Frequent ACL, Periodic ACL, Sequential ACL 그리고 Structured ACL의 관계를 도식화 한 것이다. 기존 Static ACL에서

추출한 Frequent ACL, Periodic ACL 그리고 Sequential ACL과 추출되지 않은 Static ACL이 Structured ACL을 구성하게 된다.

반복 주기와 빈도수 그리고 ACL들 간의 순서 정보를 가지는 Frequent ACL, Periodic ACL 그리고 Sequential ACL의 조합으로 구성되므로 Structured ACL 역시 해당 요소들을 포함하게 된다.

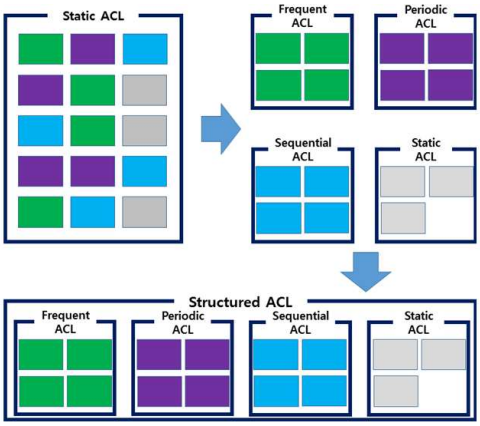

그림 1. 정의된 ACL들 간의 관계
Fig. 1. Relationship between defined ACL

2. 연관관계 추출

본 논문에서 제안한 Structured ACL은 Static ACL에 periodicity와 sequence를 추출한 순서 연관관계를 추가하여 구성된다. Structured ACL을 구성하기 위하여 Static ACL 간의 순서 연관관계를 추출하는 과정은 세 단계로 이루어진다.

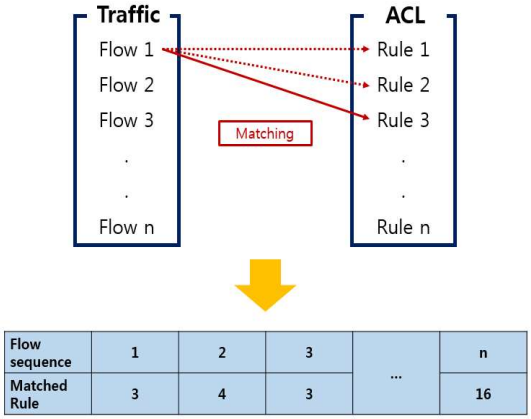

그림 2. 트래픽에 ACL을 매칭하는 과정
Fig. 2. Process of match ACL to traffic

첫 번째 단계에서는 Structured ACL 구성에 사용될 Static ACL에 각각 고유번호를 부여한다. 이는 이후 Transaction을 생성하기 위해 Static ACL을 트래픽에 매칭하여 매칭 정보를 추출하기 위해 사용된다.

두 번째 단계는 Transaction 생성 단계이다. 본 단계에서는 Static ACL과 Static ACL을 생성하는데 사용한 트래픽을 사용하여 Transaction을 생성한다. Transaction은 그림 2와 같이 트래픽을 플로우 단위로 Static ACL에 매칭시키고, 매칭되는 Static ACL의 고유 번호를 플로우의 시간 순서대로 나열한 것이다. 그림 2는 연관관계 추출의 두 번째 단계를 도식화 한 것이다. 입력 트래픽의 각 플로우에 Static ACL을 매칭하여 매칭된 ACL의 고유번호를 시간 순서대로 나열한다.

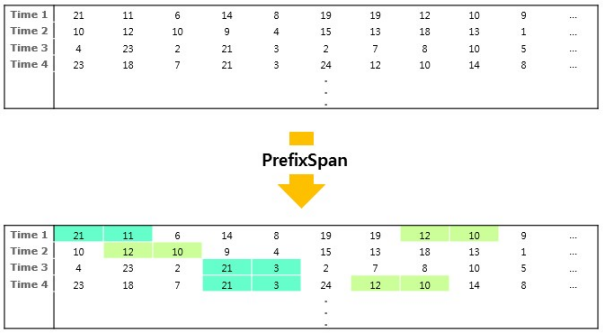

그림 3. 순서 연관관계 추출 개념
Fig. 3. Concept of sequence correlation extraction

세 번째 단계는 순서 정보 추출 단계이다. 본 단계에서는 Sequential Pattern Mining 알고리즘인 PrefixSpan 알고리즘을 사용하여 Transaction의 ACL 매칭 순서 사이의 연관관계를 추출한다. 그림 3은 PrefixSpan 알고리즘을 사용하여 Transaction으로부터 ACL 매칭 순서 사이의 연관관계를 추출하는 과정을 도식화 한 것이다. 21번 ACL 다음에는 11번 혹은 3번 ACL이 매칭되고, 12번 ACL 다음에는 10번 ACL이 매칭되는 것을 확인 할 수 있다. 그림 4는 12번 ACL 다음에 10번 ACL이 매칭된다는 순서 연관관계를 표현한 Structured ACL 예시이다. 해당 Structured ACL은 독립적인 12번과 10번 ACL을 연관관계 추출을 통하여 하나의 계층적 구조를 가지는 Structured ACL로 구성되었다. 그림 4의 Structured ACL은 계층적 구조를 통하여 12번 ACL이 매칭된 경우에만 10번 ACL을 매칭하므로 방화벽의 불필요한 탐지를 줄일 수 있다. 또한 각

ACL의 Max duration 정보를 포함하므로 해당 ACL이 매칭 된 경우에 Max duration 시간만 해당 규칙을 열어주면 되므로 모든 규칙들을 항시 열어야하는 위험을 피할 수 있다.

{(x.x.1.11, ANY <-TCP-> x.x.1.42, 382, n), (x.x.1.41, 52053 <-TCP-> x.x.1.43, ANY, n)}

그림 4. Structured ACL 모델 예시
Fig. 4. Example of structured ACL Model

3. Structured ACL 추출 시스템

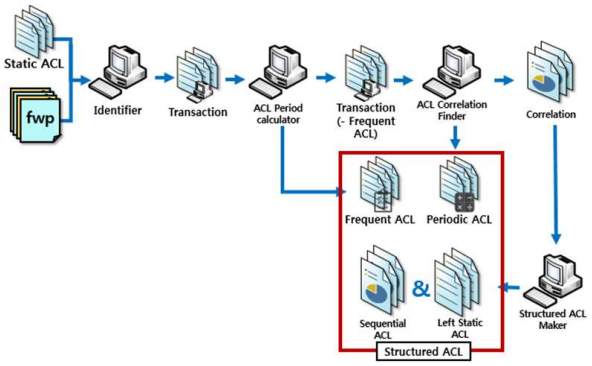

그림 5. Structured ACL 모델 추출 시스템
Fig. 5. Structured ACL model extraction system

그림 5는 Static ACL과 Static ACL을 추출하는데 사용된 트래픽을 입력으로 본 논문에서 제안하는 Structured ACL을 구성하기 위하여 Frequent ACL, Periodic ACL 그리고 Sequential ACL을 추출하는 과정을 도식화 한 것이다.

첫 번째 모듈인 Identifier는 Periodic ACL, Frequent ACL 그리고 Sequential ACL을 추출하기 위하여 Static ACL과 트래픽 파일을 매칭시켜 Transaction을 생성한다.

$$Frequency = \frac{(Total\ Input\ Time)}{(Match\ Count)}$$

그림 6. Frequency 계산식
Fig. 6. Frequency calculate formula

두 번째 모듈인 ACL Period Calculator는 1분 이하의 주기를 가지고 있는 Frequent ACL을 추출하며, 너무 빈번히 발생하는 Static ACL을 Structured ACL 구성 과정에서 제외하기 위해 사용된다. 그림 6의 수식은 총 입력 시간을 각 ACL의 매칭 횟수로 나누어 각 ACL이 얼마의 시간을 가지고 반복되는지를 계산하는 수식이다. 본 논문에서는 실험을 통해 ACL의 반복 주기가 30초 이하인 경우 빈번하다고 판단하여 해당 ACL을 연관관계 추출 과정에서 제외한다.

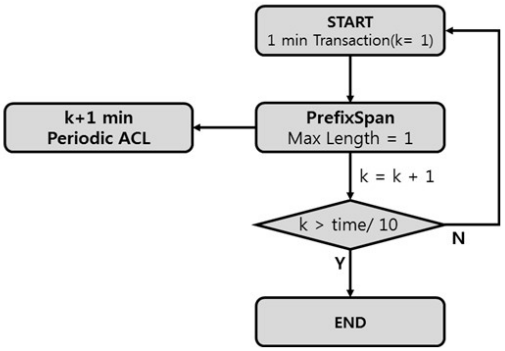

그림 7. PrefixSpan 적용을 통한 Periodic ACL 추출
Fig. 7. Periodic ACL extraction by applying the PrefixSpan

세 번째 모듈인 ACL Correlation Finder는 PrefixSpan 알고리즘을 적용하는 모듈로써 Identifier에서 생성한 Transaction에서 ACL Period Calculator에서 추출한 Frequent ACL을 제외한 내용을 입력으로 Periodic ACL을 추출한다. 그림 7은 ACL Correlation Finder에서 각 분 단위별로 반복되는 ACL을 추출하는 과정을 도식화 한 것이다. 최초로 1분 단위 Transaction을 입력으로 1분의 주기로 반복되는 Periodic ACL을 추출한다. 다시 2분 단위 Transaction을 입력으로 2분의 주기로 반복되는 Periodic ACL을 추출하며, 이 과정을 트래픽의 총 입력시간의 10분의 1이 될 때까지 반복한다. 해당 과정에서 Periodic ACL의 반복주기를 1분 단위로 설정한 것은 실험을 통해 추출된 ACL들의 반복주기가 30초를 넘어가는 경우에 대부분 1분 또는 10분 주기로 나타나는 경향이 있어, 임의로 설정하였다.

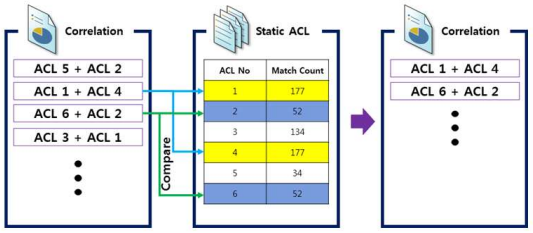

그림 8. Sequential ACL의 매칭카운트 비교
Fig. 8. Matching count comparison in Sequential ACL

네 번째 모듈인 Structured ACL Maker는 ACL

Correlation Finder를 통해 추출한 Correlation을 Match Count와 서버/클라이언트 IP의 인과관계를 기준으로 판단하여 Sequential ACL을 구성한다. 그림 8은 Match Count를 기준으로 Correlation을 판단하는 과정을 도식화 한 것이다. 첫 번째 Correlation인 ACL 5 + ACL 2는 각 ACL의 매칭 횟수가 34번과 52번으로 ACL 5번이 매칭된 이후에 반드시 ACL 2번이 매칭되었다고 볼 수 없으므로 Correlation 목록에서 제외되게 된다.

Match Count를 비교한 이후에는 Correlation으로 추출된 ACL들 간의 IP 인과관계를 확인한다. IP의 인과관계란 두 ACL들 간의 src/dst IP 비교를 통해 선행 ACL에 의해 후행 ACL이 일어났는지를 판단하는 단계이다. 본 단계에서는 선행 ACL의 src/dst IP와 후행 ACL의 src/dst IP들 간에 중복된 IP가 하나라도 있는지 확인한다. 중복된 IP가 하나 이상 있는 경우만 해당 ACL들 간에 연관관계가 존재한다고 판단한다.

최종적으로 Static ACL으로부터 추출한 Frequent ACL, Periodic ACL, Sequential ACL과 세 개의 ACL을 통해 추출되지 않은 나머지 Static ACL을 통해 Structured ACL이 구성된다.

V. 실험

본 장에서는 본 논문에서 제안한 시스템의 타당성을 증명하기 위해 실제 SCADA 네트워크 트래픽에 적용 실험한다. 표 2는 실험에 사용된 SCADA 네트워크 트래픽으로 A 사이트를 제외한 나머지 6개의 사이트 들은 모두 각 사이트에서 12시간씩 수집한 트래픽이다.

표 2는 각 사이트별로 본 논문에서 제안한 시스템을 적용한 결과를 정리한 표이다. A 사이트의 경우 전체 Static ACL의 개수는 655개이고, 추출된 Frequent ACL은 82개, 1분부터 5분까지의 Periodic ACL은 각각 3, 9, 7, 3, 5개, Sequential ACL은 30개로 중복을 제외한 ACL 원소의 개수는 18개였으며, 추출되지 않은 Static ACL은 637개이다.

표 2. 실험에 사용된 트래픽
Table 2. Test set spec

Site	Size of Traffic	No. of Flows
A	492 GB	9,198,938
B	14.6 GB	234,259
C	3.14 GB	13,729
D	10.4 GB	94,072
E	19.0 GB	1,709,279
F	2.79 GB	68,624
G	2.50 GB	637,199

실험의 결과 A 사이트 트래픽으로 추출한 총 655개의 Static ACL들 중 82개의 ACL의 Frequency가 30 이하로 추출되어 Structured ACL 추출 과정에서 제외 되었다. 또한 27개의 ACL에서 5분 이하의 반복주기를 추출 할 수 있었다. 추출된 ACL 간의 연관관계는 총 30개였으며, 중복을 제거한 고유한 ACL의 개수는 총 18개였다. 실험을 통해 전체 655개의 Static ACL들 중 18개의 ACL 간의 연관관계를 추출하여 계층적 구조를 가지는 Structured ACL로 구성할 수 있었다. 실험에 사용한 A 사이트의 경우 총 655개의 Static ACL 중 단 18개의 Static ACL들이 Sequential ACL로 구성되었지만, 해당 Static ACL들은 5분 이하의 반복주기를 가지므로 타 Static ACL들에 비하여 방화벽에서 매칭 되는 빈도수가 높다. 따라서 방화

표 3. 실험 결과
Table 3. Experiment result

Site	A	B	C	D	E	F
Static ACL	655	38	90	187	460	440
Frequent ACL	82	4	11	35	19	100
1min Periodic ACL	3	0	0	4	0	3
2min Periodic ACL	9	0	1	7	3	0
3min Periodic ACL	7	0	0	0	66	0
4min Periodic ACL	3	0	0	1	36	0
5min Periodic ACL	5	2	2	6	99	13
Sequential ACL	30(18)	2(2)	0	4(8)	3829(196)	303(16)
Left Static ACL	637	36	90	179	264	424

벽에서 1st level에서 매칭 해야 하는 ACL의 개수가 감소하고, 이에 따라 1st level에서의 search space가 줄어드는 효과를 얻을 수 있을 것으로 예상한다. 또한 ACL들 간의 순서 연관관계 추출을 통하여 제어 기기의 통신 특성을 어느 정도 반영하였다고 판단된다.

Ⅵ. 결론 및 향후 연구

본 논문에서는 Static ACL에 periodicity와 sequence를 추출한 순서 연관관계를 추가한 Structured ACL 모델을 제안하고, 이를 실제 SCADA 트래픽에 적용하여 각 규칙들 간의 순서 연관관계, Duration, S/C 정보, 반복 주기 등 다양한 정보를 포함한 Structured ACL 모델이 정상적으로 추출되는 것을 확인하였다.

본 논문에서 제안한 Structured ACL 모델이 가지는 다양한 정보들을 활용하면 기존 Static ACL 모델이 가지는 첫 번째 문제점인 Dynamic Allocated Port를 사용하는 경우에 ANY-ANY 규칙으로만 표현 가능한 문제를 해당 통신에 의해 생성된 ACL들을 발생 순서에 의해 Structured ACL로 잘 구성할 수 있다면 충분히 해결할 수 있을 것으로 생각된다. 두 번째 문제점인 제어기기의 통신 특성 반영은 ACL들 간의 발생순서 관계를 활용하여 Structured ACL을 구성함으로써 해결 하였고, 세 번째 문제점인 빈도수에 상관없이 항시 오픈해야 한다는 문제점 역시 2nd, 3rd level에 있는 규칙들은 해당 Structured ACL의 1st level에 있는 규칙이 매칭된 이후에 특정 시간동안만 오픈함으로써 해결할 수 있었다.

또한 방화벽에서 트래픽에 ACL을 매칭시에 탐색하는 1st level의 search space가 ACL들이 Structured ACL로 구성되는 비율에 따라 줄어드는 효과를 얻을 수 있을 것으로 예상된다.

향후 연구로는 플로우 내부의 비정상 탐지를 위하여 플로우 내부의 패킷 통계 정보를 모델링하여 플로우 내부의 비정상을 탐지하는 방법에 대해 연구할 계획이다.

References

[1] Yun, Jeong-Han, et al. "Burst-based anomaly detection on the DNP3 protocol." International Journal of Control and Automation 6.2 (2013): 313-324.

[2] Choi, Seungoh, et al. "Traffic-Locality-Based Creation of Flow Whitelists for SCADA Networks." Critical Infrastructure Protection IX. Springer International Publishing, 2015. 87-102.

[3] 유형욱, 윤정한, 손태식. "제어시스템 보안을 위한 whitelist 기반 이상징후 탐지 기법." 한국통신학회논문지 38.8 (2013): 641-653

[4] Lim, Y. H., Yoo, H., & Shon, T. (2013). IEC 61850 변전소 네트워크에서의이상 징후 탐지 연구. Journal of The Korea Institute of Information Security & Cryptology (JKIISC), 23(5).

[5] Schneider, Johannes, Sebastian Obermeier, and Roman Schlegel. "Cyber security maintenance for SCADA systems." Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research. British Computer Society, 2015.

[6] Igure, V.M. and Laughter, S.A. and Williams, R.D.: Security issues in SCADA networks. Computers & Security. vol. 25, no. 7, pp. 498 - 506, Elsevier (2006)

[7] Jung, Woo-suk, et al. "Whitelist representation for FTP service in SCADA system by using structured ACL model." Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE, 2016.

[8] Barbosa, Rafael Ramos Regis, Ramin Sadre, and Aiko Pras. "Flow whitelisting in SCADA networks." International journal of critical infrastructure protection 6.3 (2013): 150-158.

[9] Kang, Dong-Ho, et al. "Whitelist Generation Technique for Industrial Firewall in SCADA Networks." Frontier and Innovation in Future Computing and Communications. Springer

정 우 석 (Woo-Suk Jung)

2015년 : 고려대학교 컴퓨터정보학과 졸업
2015년 ~ 현재 : 고려대학교 컴퓨터정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

윤 정 한 (Jeong-Han Yun)

2001년 2월 KAIST 전산학과 졸업
2003년 2월 KAIST 전산학과 석사
2011년 2월 KAIST 전산학과 박사
2010년 12월~현재 ETRI부설국가보안기술연구소 선임연구원
<관심분야> 프로그램 분석, 제어시스템 네트워크 침입탐지

김 신 규 (Sin-Kyu Kim)

2000년 2월: 연세대학교 기계전자공학부 졸업
2002년 2월: 연세대학교 컴퓨터과학과 석사
2014년 2월: 연세대학교 컴퓨터과학과 박사
2003년 12월~현재: ETRI부설국가보안기술연구소 선임연구원/실장
<관심분야> 스마트그리드 보안, 국가기반시설 보안, 취약점 분석

구 영 훈 (Young-Hoon Goo)

2016년 : 고려대학교 컴퓨터정보학과 학사
2016년~현재 : 고려대학교 컴퓨터정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

이 성 호 (Sung-Ho Lee)

2016년~현재 : 고려대학교 컴퓨터 정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 트래픽 분류

김 명 섭 (Myung-Sup Kim)

1998년 : 포항공과대학교 전자계산학과 학사
2000년 : 포항공과대학교 전자계산학과 석사
2004년 : 포항공과대학교 전자계산학과 박사
2006년: Dept. of ECS, Univ of Toronto Canada
2006년~현재 : 고려대학교 컴퓨터정보학과 교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크