Journal of Convergence for Information Technology
Vol. 9. No. 8, pp. 35-44, 2019
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2019.9.8.035

상용 OS기반 제어시스템 확률론적 취약점 평가 방안 연구

엄익채
한전KDN(주) 보안컨설팅팀 차장

A Study on the Probabilistic Vulnerability Assessment of COTS O/S based I&C System

Ieck-Chae Euom
Deputy General Manager, Cyber Security Consulting Team, KEPCO KDN

요 약 본 연구는 즉시 패치가 어려운 상용 운영체제 기반의 계측제어시스템의 취약점 평가 방안 및 시간의 경과에 따른 위험의 크기를 정량적으로 파악하는 것이다. 연구 대상은 상용 OS가 탑재된 계측제어시스템의 취약점 발견과 영향의 크기이다. 연구에서는 즉각 취약점 조치가 힘든 디지털 계측제어시스템의 취약점 분석 및 조치방법을 연구함으로써, 계측제어시스템이 존재하는 핵심기반시설의 전체적인 사이버보안 위험과 취약점을 정량적으로 파악하는 것이다. 본 연구에서 제안한 확률론적 취약점 평가 방안은 즉각적인 취약점 패치가 어려운 상용 운영체제 기반의 계측제어시스템에서 취약점 패치 우선 순위 및 패치가 불 가능시 수용 가능한 취약점의 임계값 설정, 공격 경로에 대한 파악을 가능하게 하는 모델링 방안을 제시한다.

주제어 : 핵심기반시설, 위험 모델링, 취약점 생명주기, 취약점 발견모델, 공격 그래프, 마르코프 모델

Abstract The purpose of this study is to find out quantitative vulnerability assessment about COTS(Commercial Off The Shelf) O/S based I&C System. This paper analyzed vulnerability's lifecycle and it's impact. this paper is to develop a quantitative assessment of overall cyber security risks and vulnerabilities I&C System by studying the vulnerability analysis and prediction method. The probabilistic vulnerability assessment method proposed in this study suggests a modeling method that enables setting priority of patches, threshold setting of vulnerable size, and attack path in a commercial OS-based measurement control system that is difficult to patch an immediate vulnerability.

Key Words : Critical Infrastructure, Risk Modeling, Vulnerability Life Cycle, Vulnerability Detection Model, Attack Graph, Markov Model

1. 서론

2014년 국내에서 발생한 한국수력원자력 해킹사건으로 인해 원자력발전소 보안 위협에 대한 인식이 높아져 있다. 망 분리 등 최신 보안기술이 적용되었던 한수원의 원자력 관련 내부 자료들이 해킹되어 당시 사회적 이슈로 대두되었으며, 북한과 대치중인 우리나라에 있어서 안보 관련 위협이 구체화, 현실화된 사례이기도 하다[1].

최근까지, 국내 원자력 발전소의 보안 기술은 외부 경계 보호에만 치중하는 실정이나, 최근 4차 산업혁명의 부각으로 외부와의 통신 필요성이 증가하고 있으며 관련된 보안 취약점의 우려도 커지고 있다. 특히 2017년 랜섬웨어 공격으로 인해 PLC(Programmable Logic Controller)를 포함한 제어시스템등도 랜섬웨어에 감염될 수 있다는 것이 증명[2]되고 있어, 취약점에 의한 국가 기반시설의 위험성이 우려할

*Corresponding Author : Ieck-Chae Euom(icelaken@gmail.com)
Received July 18, 2019
Revised August 7, 2019
Accepted August 20, 2019
Published August 28, 2019

만한 수준이다.

최근 도입되는 계측제어시스템은 디지털화되어있으며 Windows등의 상용 운영체제가 설치된 시스템들로 구성되어 있다. 이러한 상용 운영체제가 설치된 계측제어 시스템 상의 알려진 취약점들이 시스템의 생애 주기 동안 어떤 특성으로 전파되고 사전 예방할 수 있는지에 대한 연구가 필요하다. 특히, 원자력 발전소의 경우 18개월마다 시행하는 핵연료 교체 시기 이외에는, 발전소 안정성 때문에 취약점 패치 등을 시행할 수가 없다. 또한 외부 인터넷과 네트워크 연결이 단절되어있어 네트워크를 통한 취약점 공격이 불가능한 면도 있다. 이러한 원자력 발전소등 핵심기반시설의 특성을 감안한 계측제어시스템에 대해 취약점을 패치하지 못할 때 시간 흐름에 따라 어떠한 영향력 크기를 가지는지를 연구하는 확률론적 평가 모델링에 대한 연구가 필요하다.

본 논문의 연구목적은 즉각 취약점 패치가 힘든 발전소 등의 계측제어시스템의 취약점 분석 및 예측방안을 연구하여, 핵심기반시설에 대한 취약점의 영향력을 정량적으로 파악하는 것이다.

본 연구에서는 계측제어시스템에 존재하는 취약점에 대해 침해 가능한 공격 경로를 파악하고, 시간 경과에 따른 취약점 영향을 확률 기반 및 정량적인 방법으로 평가하는 방법을 제안하고자 한다. 이를 통하여 계측제어 시스템 내에 존재하는 취약점에 대해 동적이고 정량적인 위험 지수를 산출한 후, 패치하지 못하는 취약점이 가지는 위험도를 동적으로 산출한다. 또한 계측제어시스템 내 존재하는 여러 취약점의 패치 우선순위를 결정한다.

2. 관련 연구

일반 IT인프라 에서는, 취약점을 발견되면 이를 즉시 패치 하는 것이 보안 조치의 중요한 활동 중의 하나이다. 하지만 계측제어시스템 등이 포함되어 있는 핵심기반시설에서는, 신규 취약점 발견과 패치 적용활동이 보안 조치활동 중 큰 부분을 차지하지 못하고 있다. 핵심기반시설에서는 시스템의 안정성이 최우선 순위이기 때문에, 취약점을 적시에 패치하는 것은 현실적으로 힘든 일이다.

이런 핵심기반시설의 제어시스템 취약점 패치에 대해 미국의 가이드라인을 살펴보면, Fig. 1은 미국 국토안보부가 권장하는 제어시스템의 보안 취약점 패치 가이드라인이다[3]. 이 절차는 아래 3가지 상황 판단에 따라 패치 시행 여부 및 시기를 결정한다.

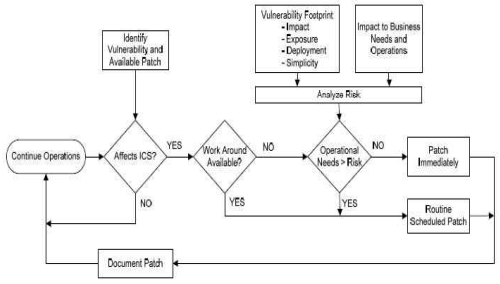

Fig. 1. Vulnerability Patch Process by DHS

첫 번째, 취약점이 제어시스템에 악영향을 끼치는지 파악한다. 해당 취약점이 감염되었을때 악영향을 끼치지 않는다면 패치를 시행하지 않아도 되고, 영향을 끼친다면 아래 의사 결정 사항을 고려한다.

두 번째, 해당 제어시스템에 대한 주기적 감시가 가능하면 발전소 정기 유지보수 기간에 취약점 패치를 시행한다. 만약 주기적인 감시가 불가능하면 아래 의사 결정 사항을 고려한다.

세 번째, 해당 제어시스템에 취약점의 감염 가능여부와 발생 가능한 침해 크기를 파악하여 위험 분석을 실시한 후, 산출된 위험이 제어시스템 지속적 운영으로 인한 이익보다 크면 즉시 취약점 패치를 실시한다.

이런 제어시스템의 취약점 패치 프로세스는 시간의 경과에 따른 취약점의 영향력 변화를 고려하지 않았고, 취약점 패치를 하지 않을 경우 발생하는 위험 크기를 정량적으로 계산하기 힘든 단점이 존재한다. 핵심기반시설의 현장 제약사항을 고려하여, 계측제어시스템 취약점 패치를 효과적으로 수행하기 위해서는 먼저 취약점의 생명주기를 분석하고, 침해 가능한 공격 경로 상에 존재하는 취약점들의 시간의 경과에 따른 영향을 파악하는 확률론적 취약점 평가 모델링이 필요하며 정량적 보안 지수를 산출하여야 한다.

3. 연구내용

3.1 연구대상 및 범위

첫째, 연구에서 중점을 두는 계측제어시스템의 범위이다. Fig. 2는 계측제어시스템의 구성을 보여주고 있다[4]. 계측제어시스템의 목적은, 현장설비의 데이터를 계측(Instrumentation)하고 제어(Control)하는 것이다. 계측 제어시스템은 발전소 운영자가 직접 플랜트 설비를 제어하는 HMI(Human Machine Interface), 수 많은 제어설비들이 연결되어 다중화 된 분산제어시스템(Distributed Control System), 현장 설비들을 제어

하는 PLC (Progra mmable Logic Controller)로 구성된다.

본 연구는 HMI 와 DCS 에 존재하고 있는 상용 OS가 설치된 시스템을 대상으로 연구하였다. 그 이유는 운영자를 포함한 외부에 노출된 시스템이 상용 OS가 설치된 워크스테이션에 국한된 계측제어시스템의 특성 때문이며, 대부분 계측제어시스템은 외부 네트워크와 단절되어 있어 외부 공격자에게 침해가 가능한 구간은 HMI구간 외에는 없는 점을 고려하였다.

Fig. 2는 발전소에서 운영 중인 계측제어시스템을 나타내고 있다. 이 계측제어시스템에서 현장 핵심 제어 설비(Turbine, Generator)를 침해할 수 있는 경로는 M4 장비인 Controller외에는 없다는 것을 볼 수 있다. Fig.2에서 보면 운영자나 유지보수 관리자가 접근 가능한 시스템은 M1 장비인 HMI이다. 즉 Controller를 침해하기 위한 노드 상의 경로는 (M1→M4),(M1→M2→M4),(M1→M3→M4),(M1→M2→M3→M4)으로 파악할 수 있다. 이는 물리적인 연결 경로로만 파악한 경우이고, 실제 시스템간에 연계된 서비스 포트 등을 감안할 경우 다양한 공격 경로를 파악할 수 있다.

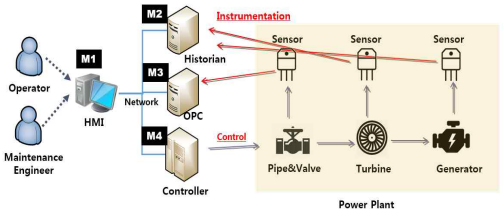

Fig. 2. Structure of Digital I&C System

둘째, 정량적인 취약점 평가를 위해서는 정량적인 보안지수의 이용이 필요하다. Pubudu et.al의 연구에서는 정량적인 보안지수의 종류는 Fig. 3과 같이 구분하고 있다[5].

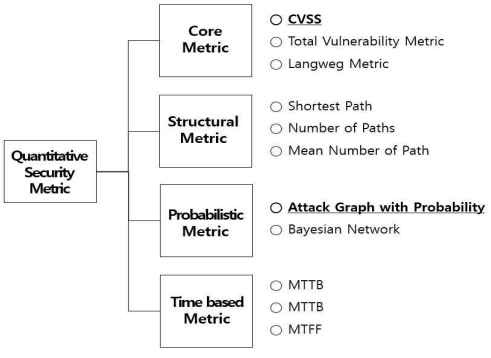

Fig. 3. Classification of Quantitative Security Metric

정량적 보안지수의 기본은 기존 공통취약점평가체계 중, 기본 수치(Base metric)를 산출하는 공격 벡터, 공격 복잡도, 침해 시 영향 크기(무결성, 기밀성, 가용성), 인증여부 등을 사용한다[6]. 본 연구에서는 상용 운영체제가 설치된 계측제어시스템이 연구대상이므로, 상용 운영체제의 취약점 지수 평가 방식으로 가장 널리 사용 중인 CVSS를 선정하였으며, 시스템에 존재하는 취약점의 CVSS스코어링 점수를 공격 그래프 노드상에 확률론적으로 반영하는 확률기반 공격그래프(Attack Graph with Probability)를 중심으로 연구한다.

마지막으로, 시간의 경과에 따른 예측 가능한 취약점 평가 모델링을 위해, 기존 학계에서 연구되고 있는 취약점 발견 모델과 취약점 생명주기 모델을 고려한다. 이 두 가지 모델을 시간을 중심축을 하여 확률 기반 공격그래프에 적용하여, 시간 경과에 따라 공격그래프 상 노드 간에 취약점의 전이 확률을 동적으로 계산하는 방안을 연구하였다.

3.2 연구방향

핵심기반시설의 포함된 산업 제어시스템(ICS)중, 특히 원자력 발전소등에서는 발전소 설비의 안전성 평가 등에서 확률론적 평가방법을 오래전부터 광범위하게 사용하고 있다[7]. 이러한 기법을 계측제어시스템의 취약점 평가에도 접목해 보는 것이 본 연구의 시작점이다.

이러한 확률론적 평가 모델링을 취약점 평가에 접목해서 얻을 수 있는 장점은 신규 취약점이 발견되었을 경우, 취약점 패치를 못 하고 시간이 경과될 때 취약점의 동적 위험 크기 등을 정량적으로 예측할 수 있는 것이다. 이런 확률론적 평가 모델링 기법이, 기존 안정성 평가 등에서 사용되고 있지만 취약점 평가와는 차이가 있다. 취약점 평가 분야에서는 실패(Fail)이라고 규정할 수 있는 사이버 침해가 외부자 또는 취약점 감염으로 인한 의도적인 사고로 볼 수 있지만, 기존 안전성 평가 측면에서는 이러한 실패가 랜덤한 확률로 발생할 수 있는 사고로 볼 수 있다. 즉 확률론적 평가 관점에서는 발생 가능한 확률에 대해 바라보는 관점이 다르다.

공격 그래프는 네트워크로 연결되어 있는 시스템에서 공격의 전이를 보여주는 강력한 평가 모델링 기법[8] 중의 하나이다. 이런 공격 그래프의 노드 간의 공격 전이 가능성에 확률론적 모델링을 응용할 경우 정량적이고 예측 가능한 취약점 평가를 할 수 있을 것이다. 이러한 가정 하에서 본 연구에서는 공격 그래프상의 노드간의 전이를 마르코프 모델을 기본으로 한 확률론적 보안 평가 모델링방안을 제안한다.

마르코프 모델은 IT시스템분야 등에서 성능 측정 및 의존성 분석 등에서 사용되는 기술이다. 마르코프 프로세스는 소위 "마르코프 상태"(Markov property)를 충족하는 프로세스를 의미한다. 마르코프 상태는 미래의 상태가 과거 상태에 의존하지 않고, 단지 현재 상태에만 의존한다는 개념이다[9]. 이러한 마르코프 모델을 공격그래프 등에 포함된 네트워크에 응용한 것을 마르코프 체인 모델이라고 한다. 이러한 마르코프 체인 모델은 상태 간 전이 확률을 기본으로 다수 상태를 보유하고 있는 시스템의 특정 상태에 존재할 확률을 계산하는 방법이다. 이는 시스템 신뢰도등을 분석하는데 많이 사용되고 있다. 확률론적 분석 과정으로 마르코프 과정에서 사용되는 마르코프 체인은 아래와 같다[10].

마르코프 체인에서 X(t)가 확률 과정 상태일 때, 어떤 n이 X(n)가 취하는 값을 상태라고 한다. 이 때 t가 특정한 시간 X(n) = a_i인 것을 n단계 시행에서 상태 a_i에 있다고 부른다. 따라서 확률변수 X(t)가 $X_1,X_2,X_3,...,X_n$일 때 대응하는 상태 공간은 유한가산 집합 $a_1,a_2,a_3,...,a_n$ 에 속한다. 상태의 순서쌍 (a_i,a_j)에 존재할 확률은 $P_{ij}(n)$으로 나타내며, 확률변수 X(n),X(n+1)의 상태가 a_i,a_j 일 경우 상태 a_i의 바로 다음 단계의 시행 결과의 확률은 $P_{ij}(n)$이라고 부른다. 마르코프 체인에서는, 측정할 수 있는 상태들의 집합을 S라고 한다. $S=(S_1,S_2,S_3..,S_r)$라고 할 때, 시간에 관계없이 동일한 결과를 가질 경우 이를 이산 시간 마르코프 체인[11]이라고 부른다. 이러한 확률적 특성을 아래 수식과 같이 정리한다.

$$P(X_k=S_k,X_{k-1}=S_{k-1},X_{k-2}=S_{k-2},.)=P(X_k=S_k|X_{k-1}=S_{k-1}),\ S_j \in S$$

이산 시간 마르코프 체인에서는, 특정 상태에서 다른 상태로 이동을 하는 것을 전이(transition)라고 한다. 예를 들어 S_i상태에서 S_j상태로 이동하는 확률을 P_{ij}라고 한다. 이 확률값은 그 이전의 상태에 영향을 받지 않으며 이런 확률 P_{ij}를 전이확률이라고 부른다[12]. 이런 특성을 가진 마르코프 체인을 시간 동질성(time homogeneous)을 가졌다고 하며 아래 수식과 같이 표기한다[13].

$$P(X_k=j|X_{k-1}=i)$$

마르코프 체인에서 전이 행렬이 나타내는 것은 노드가 특정 상태에서 다른 확률론적 상태로 전이하는 확률이다. 마르코프 체인에서 매개변수들은 전이 행렬로 나타낼 수 있다. Fig. 4는 전이 행렬(transition matrix)의 구조를 보여준다[14].

$$P=\begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,m} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,m} \\ \cdot & \cdot & \cdot & \cdot \\ P_{n,1} & P_{n,2} & \cdots & P_{n,m} \end{pmatrix}$$

Fig. 4. Transition Matrix

이러한 추이 과정인 마르코프 체인을 확률론적 취약점 평가에 적용하기 위해서는 공격자의 행위를 분석해야한다. 단일 시스템에 여러 개의 취약점이 존재할 수 있는데, 본 연구에서 제안하는 모델에서는 공격자가 최종 목표를 침해할 때, 가장 익스플로잇 가능성이 큰 취약점을 이용한다는 가정 하에 모델링을 하였다.

4. 제안하는 확률론적 취약점 평가 모델링

4.1 제안 프레임워크

본 연구를 통해 제안하는 모델링의 프레임워크는 Fig. 5와 같다. 본 연구의 목적인 정량적인 취약점 평가 모델링과, 취약점이 시간 경과에 의한 위험도 변화를 나타내는 동적인 모델링을 구현하기 위해 Fig. 5와 같은 프레임워크를 제안한다. 프레임워크는 두 개의 축을 가지며, X축은 취약점 분석 방향을 Y축은 정량적 보안 지표의 성격을 나타낸다.

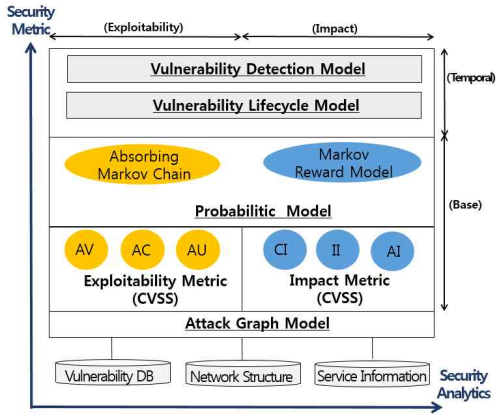

Fig. 5. Proposed Probabilistic Vulnerability Assessment Framework

X축은 취약점 분석 방향을 의미하며, 좌측은 취약점의 위험도(Exploitability)를 나타낸다. 위험도 지표는 세 가지 지표로 구성이 되어 있는데, 공격 벡터를 나타내는 AV (Access Vector) 공격 복잡도를 나타내는 AC (Access Complexity)

인증 복잡도를 나타내는 AU (Authentication)로 구성된다. 오른쪽은 취약점으로 인한 피해(Impact)크기 측면으로 분석을 한다. 피해 크기 지표 도 세 가지로 구성이 되는데, 기밀성의 침해 크기를 표현하는 CI(Confidentiality Impact) 무결성 침해 크기를 표현하는 II(Integrity Impact) 가용성 침해 크기를 표현하는 AI(Availability Impact)로 구성이 된다[15].

취약점의 위험도 측면을 분석하기 위해, 취약점이 공격 그래프의 공격 경로에서 시간 경과에 따른 노드간의 전이 확률을 마르코프 체인으로 구현하였다. 취약점으로 인한 피해 크기 측면은 마르코프 보상 모델[16]을 적용하였다. 이처럼 마르코프 체인과 마르코프 보상 모델을 적용하여 취약점 위험도 지표와 취약점 위험 영향지표를 산출한다. 취약점 영향 크기를 위험도 측면과 피해 정도 측면으로 구별하는 것은 CVSS의 기본 점수(Base Metric)산출에서도 사용하는 방식이다. 다만 본 연구에서는 이런 부분을 공격 그래프상 노드 간의 이동이 나타내는 마르코프 성질을 통해, 마르코프 체인과 더불어 마르코프 보상 모델을 적용하였다.

Y축은 총 5개 분야로 구성되어 있다. 아래 3개 분야는 네트워크 시스템이 가지는 본질적인 취약점의 특성을 분석하는데 목표가 있으며, CVSS에서 가장 본질적 지표인 기본 지표(Base Metric)를 사용한다. 위 2개의 분야는 시간 경과에 따른 취약점의 동적 특성을 나타낸다. 시간경과에 따른 취약점의 영향 크기를 반영하는 취약점 생명 주기 모델과, 시간 경과에 따른 취약점 발견율을 반영하는, 취약점 발견모델을 제안한다.

첫 번째 분야는 공격 그래프를 생성하는 분야로, 공격 그래프를 생성하는데 필요한 취약점DB, 네트워크 연결 구성 정보, 노드에 설치된 서비스 정보를 사용한다. 두 번째 분야는 공격 그래프 노드 상에서 발견된 취약점들을 위험도 지표와 위험 영향 지표를 보여주는 여섯개의 세부 속성으로 나타낸다. 세 번째 분야는 공격 그래프상의 노드 간 전이를 확률론적으로 나타낸 흡수 마르코프 체인(Absorbing Markov Chain)과, 공격 그래프상 노드들의 침해 영향을 정량적으로 나타내는 마르코프 보상 모델(Markov Reward Model)을 사용한다. 네 번째 분야는 시간 경과에 의한 취약점의 위험도 지표를 동적으로 계산하는 취약점 생명 주기 모델을 사용하여 공격 그래프 노드 간의 전이 확률에 시간을 반영한 동적인 모델링을 제안한다. 다섯 번째 분야는 향후 발견될 취약점을 예측하는 분야로, 특정 소프트웨어가 시간의 흐름에 따라 발생 하는 취약점 수를 예측하는 취약점 발견 모델을 제안한다.

본 연구에서는 세 번째 분야로부터 다섯 번째 분야를 동일한 시간 축에서 확률론적으로 노드 간 전이 확률을 예측 및 계산 할 수 있는 통합된 방안을 제시한다.

4.2 제안 모델링 프로세스

본 연구에서 제안하는 프레임워크의 구현 프로세스를 나타내면 Fig. 6과 같으며, 총 네 단계 프로세스로 구성되어 있다.

첫 번째 프로세스는, 공격 그래프 생성(Attack Graph Generation)이다. 공격 그래프 생성에는, 네트워크 구성 토폴로지, 각 노드에 운용 중인 서비스를 고려하여야 한다. 운용 중인 서비스 별로 사용하는 네트워크 포트 및 어플리케이션이 구분되며, 존재하는 취약점이 구분되기 때문이다. 공격 그래프를 만드는 방법은 여러 가지가 있으며, 본 연구에서는 “MulVAL”이라는 오픈소스 기반 공격 그래프 생성 도구를 제안한다.

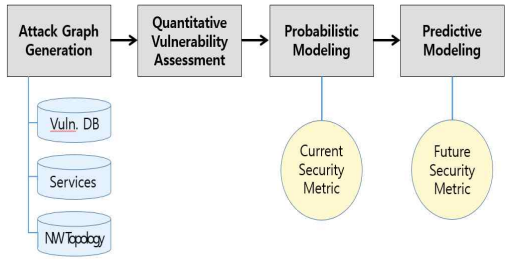

Fig. 6. Proposed Probabilistic Vulnerability Assessment Process

두 번째 프로세스는 정량적 취약점 평가(Quantitative Vulnerability Assessment)이다. 본 단계에서는 공격 그래프의 노드, 즉 호스트의 취약점 점수를 세부 요소로 구별하여 분석한다. 크게 두 가지 요소로 구분하여 분석을 하는데, 취약점 위험도(Exploitability)와 영향 크기(Impact)이다. 취약점 위험도 분석 측면은, 공격자가 해당 시스템을 침해하는데 필요한 노력과 소요되는 시간을 파악한다. 이를 반대로 보면 해당 시스템이 공격자에게 얼마나 강력하게 안전한지를 나타낸다. 취약점 영향 크기 측면은, 침해 시 발생하는 시스템의 피해를 분석한다.

공격자에 의해 쉽게 감염 및 전파가 가능한 취약점의 경우, 이로 인한 침해 피해 크기는 상당히 클 것이다. 이러한 점을 고려하여, 취약점의 위험도와 취약점 영향 크기는 상관관계가 없다는 것을 전제 할 수 있다.

또한, 각 취약점의 세부 요소들도 분석을 하여야 한다. 취약

점 위험 측면의 요소인 Access Vector, Access Complexity, Authentication 요소를 고려하고, 취약점의 피해 크기 측면에서는 기밀성, 무결성, 가용성 측면에서 분석을 한다.

세 번째 프로세스는 확률론적 모델링(Probabilistic modeling)이다. 확률론적 모델링 단계에서는, 공격 그래프 상 노드들의 취약점들의 관계를 분석해본다. 확률론적 모델링 단계에서는 상태 전이 다이어그램상에서 시간 경과에 의한 확률론적 특성을 고려하여 분석을 한다. 본 연구에서는 시스템의 정량적 취약점 특성을 나타내기 위해 흡수 마르코프 체인 모델의 사용을 제안하였다.

시스템 취약점의 정량적 지수 산출을 위해서. 기대 경로길이(expected path length), 확률론적 경로 길이(probability path metric), 노드 순위(node ranking)등의 수치를 제안한다. 이런 정량적 지수들은 취약점을 감염시키는 시간과 노력의 크기를 나타낸다.

마지막 프로세스는 예측 모델링(Predictive modeling)이다. 취약점이 가지는 동적이고 임시적 특성을 고려하여, 본 연구에서는 취약점 생명주기 모델을 제안하였다. 취약점 생명주기 모델링의 목적은 시간의 경과에 따른 취약점의 특성을 파악하는 것이다. 또한 소프트웨어 생명주기 동안 향후 발견될 취약점의 발견에 대해 예측이 필요하며, 이는 취약점 발견 모델링을 통해서 예측할수 있다. 취약점 발견 모델링을 이용하면 특정 소프트웨어의 시간 경과에 따른 취약점 발견 빈도수를 추정할 수 있다.

Fig. 7은 이러한 예측 모델링의 전체적인 프로세스를 정리해서 보여주고 있다. 취약점 생명주기 모델 중에서는 가장 범용적으로 사용하고 있는 Frei 모델[17]을, 취약점 발견 모델중에서는 Windows 운영체제에 대한 통계학적 데이터가 풍부한 AML모델[18]을 사용하였다.

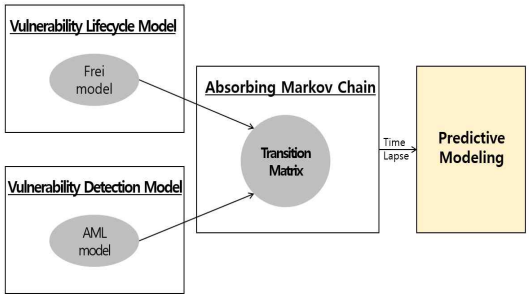

Fig. 7. Proposed Predictive modeling Process

4.3 제안 모델링 알고리즘

현재까지 살펴본 모델링 프로세스를 의사(pseudo) 알고리즘으로 나타내면 Fig. 8과 같다.

```
1: Perform Vulnerability Scan of network N
2: for each vulnerability  x ∈ V
3:          extract CVE-ID
4:          extract CVSS Description from CVE DB
5:          extract CVSS Vectors (AV,AC,AU) from NVD
6: Generate AttackGraph()
7: Assign Normalized Probabilities to transitions
   between  Nodes in Attack Graph
8: Apply Stochastic Model
9: Generate Security Metrics
```

Fig. 8. Proposed modeling's pseudo algorithm

의사 알고리즘별 라인의 의미는 아래와 같다.

(1) 계측제어시스템의 IP기반 네트워크에 대한 취약점 스캐닝을 시행한다. 취약점 스캐너는 Nessus등의 오픈소스 기반의 취약점 스캐너를 사용한다.
(2) 발견된 취약점을 호스트 단위로 구분한다.
(3) 취약점에 해당하는 CVE-ID를 추출한다.
(4) CVE DB 나 ICS-CERT에서 관련된 취약점 정보를 확인한다.
(5) 해당 취약점의 CVSS스코어링 점수에서 세부 요소 점수를 추출한다. Access Vector, Access Complexity, Authentication 점수를 추출한다.
(6) 공격 그래프를 생성한다. 공격 그래프 생성 시 취약점 정보, 서비스 정보, 네트워크 토폴로지 정보 등을 이용한다.
(7) 공격 그래프 상 노드 간의 전이 확률을 반영한다.
(8) 확률론적 모델을 적용한다.
(9) 정량적 보안지수를 산출한다.

본 연구에서 가장 기본적으로 사용하고 있는 공통 취약점 평가 체계도 한계점은 존재한다. 첫째, 기본 지수(Base metric)산정 단계에서 위험도 수치와 위험 영향 수치 사이에 상관 관계가 없다는 점이다. 또한 공통 취약점 평가체계에서 사용되는 속성값들이 현장의 취약점 정보, 특히 핵심기반시설의 제어시스템 취약점 정보를 반영하기에는 한계가 있다. 이러한 제약 사항은 공통 취약점 평가체계 사용이 취약점에 대한 잘못된 판단 결과를 가져온다는 것을 의미한다. 기본적으로 취약점은 여러 가지의 악용(exploit) 시나리오를 가지고 있고, 어떤 시나리오가 실행되는가에 따라 취약

점으로 인한 피해가 달라질 수 있다. 그렇다고 공통취약점 평가체계가 무용지물은 아니며, 취약점을 평가하는 하나의 기준으로 삼기에 충분하다.

앞서, 공통 취약점 평가체계가 핵심 기반 시설의 취약점 정보를 반영하기에는 한계가 있다는 점을 언급하였다. 예를 들면, 계측제어시스템에서 발견된 취약점이 낮은 공통취약점 평가체계 점수이더라도, 네트워크상에 연계된 다른 노드들의 취약점들과 결합되었을 경우에는 큰 침해 크기를 나타낼 수 있다. 이러한 한계점을 보완하기 위해 공격 그래프상의 노드에 존재하는 취약점의 공통 취약점 평가체계 점수와 세부 속성값을 공격 그래프에 반영하여, 취약점의 심각도를 확률론적으로 평가하는 방안을 제안한다.

확률론적 모델 적용에 있어서 첫 번째, 공격 그래프를 마르코프 체인에 적용한다. 공격 그래프는 마르코프 체인과 비슷한 형태를 가지고 있으며, 특히 공격 그래프상에서 최종 노드(root node)인 침해 상태 노드에서는 다른 노드로 전이가 발생할 필요가 없다. 이는 마르코프 체인 종류 중 한 가지인 흡수 마르코프 체인과 동일한 형태를 나타낸다. 그래서 본 연구에서는 작성한 공격 그래프를 흡수 마르코프 체인으로 변환하여 노드 간의 취약점 전이를 확률론적으로 모델링한다. 확률론적 모델링 단계에서는 시간 경과에 상관없이 개별 취약점의 위험성 지수(exploitability score)를 바탕으로 한 전이 확률에 기반을 두고 있으며, 공통 취약점 평가체계의 시간 지수 영역은 취약점의 생명주기 동안의 영향도를 반영하고 있다. Fig. 9.는 예측 모델링 프로세스(Predictive modeling process)를 보여주고 있다. 공격 그래프 노드에 존재하는 취약점의 전이는 해당 취약점의 위험성 지수(exploitability score)에 따라 달라진다. 또한 시간의 경과를 반영하기 위해, 취약점 생명주기와 취약점 발견 모델을 적용, 노드간의 동적인 전이확률을 산출하여 공격 그래프상에 표현한다. 이러한 과정을 통하여 시간축에 일관성 있는 취약점 예측 모델링 결과를 보여줄 수 있다.

예를 들면 CVE-2014-3513 이라는 취약점의 경우를 살펴본다. 이 취약점은 오픈소스인 OpenSSL의 소프트웨어 약점을 이용하여 서비스 거부 공격을 일으키는 취약점이다. 공통취약점 평가체계에 의해 등록된 취약점의 기본 위험 지수는 8.6으로서, 고위험(critical)에 해당하는 취약점이다. 이 취약점은 2015년 1월에는 익스플로잇 가능한 공격 코드가 존재하지 않아, 위험 상태가 "비 증명(Unproven)"이었다. 공통 취약점 평가체계에서, "비 증명 상태"인 취약점의 임시 가중치는 0.85이다. 이런 가중치 정보를 기반으로 한 실제적인 감염 지수는 아래 수식으로 표현된다.

$$e(v_i) = \text{임시 가중치} \times e(v)$$

이 수식에 적용하면 CVE-2014-3513의 임시적 감염 지수는 7.31이 된다. 취약점이 발견된 지 오래되었고 익스플로잇 가능한 코드가 공개 되면 감염 지수는 기본 감염 점수와 비슷하게 변한다. 기본 감염 점수가 같은 두 가지의 취약점을 예를 들어 본다.

"CVE-2019-0551"과 "CVE-2019-3513" 취약점의 기본 감염 점수는 8.6이다. 하지만 2019년 5월 1일 기준으로 볼때 "CVE-2012-0551"은 익스플로잇 가능 코드가 공개된 상태이며, 감염 상태는 "고(high)"로 나타난다. 그래서 임시 가중치 스코어는 1이 되며, 이를 통해 효과적인 감염 지수는 8.6으로 산출이 된다. 이 비교를 통해 두 가지 취약점의 기본 위험성 지수는 같지만, 2019년 5월 1일 시점에서는 "CVE-2012-0551"취약점이 이 더 위험하다는 것을 판단할 수 있다.

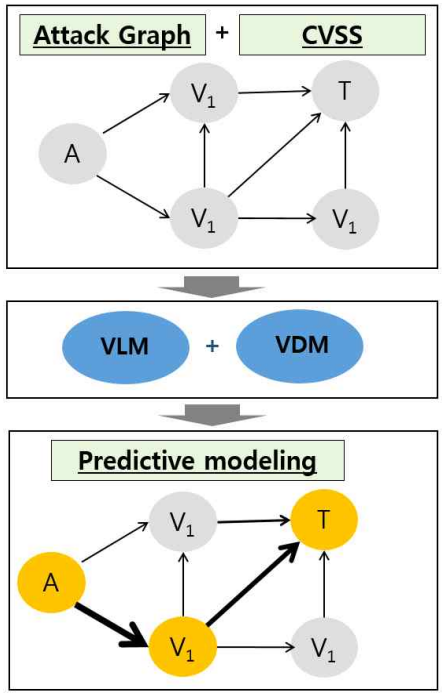

Fig. 9. Predictive modeling process

4.4 제안 모델링 개념에 대한 검증

연구에서 제안한 모델링 알고리즘에 대해, 실제 테스트베드에서 시험하기 전에, 개념 차원의 검증을 해 본다.

Fig. 2의 계측제어시스템을 대상으로, 공격 그래프에서 취약점 발견 모델이 어떻게 통합되는지 개념적으로 실험을 해 보았다. Fig. 10은 Fig. 2가 보여주는 계측제어시스템의 공격그래프 일부를 개념적으로 보여준다.

본 그래프에서 V_i는 패치되지 못한 취약점을 나타낸다. 본 연구에서는 공격 그래프와 취약점 발견 모델링을 통합하는 것에 중점을 두기 때문에, 공격그래프상 노드에 존재하는 네 가지의 취약점의 감염 지수는 같다고 가정한다.

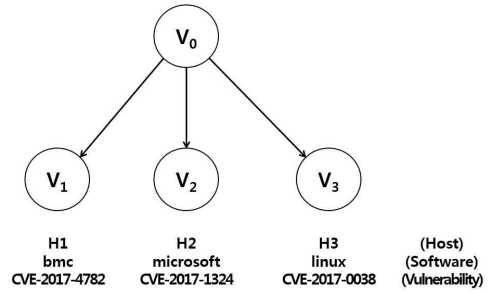

Fig. 10. Case-Initial Attack Graph

위 공격그래프 노드 상에 존재하는 소프트웨어의 취약점 발견율은 모두 다르고, 특정 시간 t 이후에 발견되는 취약점의 수는 노드에 존재하는 소프트웨어에 따라 달라진다. 취약점 발견 모델링 기법 중의 하나인 AML모델에서는, 소프트웨어의 누적된 취약점 수는, 소프트웨어 시장 점유율, 발견되지 않은 취약점 수, 사용율에 의해 결정된다. Fig.11은 특정 시간 t 이후에, 공격 그래프 노드에서 새롭게 발견되는 취약점들을 개념적으로 보여준다.

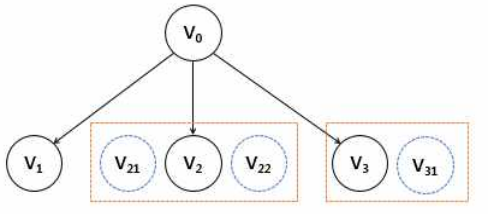

Fig. 11. Case1-Attack Graph combined with VDM

특정 시간 t이후, 신규 발견된 취약점은 (V_{21},V_{22},V_{31})이다. 이렇게 가정을 한 근거는, H2에 설치된 소프트웨어제조사인 Microsoft가 3개 소프트웨어 중에서 가장 시장 점유율이 높고, 활성화되어 있기 때문이다. 그래서 H2 노드에는 두 가지 신규 취약점 V_{21},V_{22}이 존재한다. Linux가 설치된 H3는 V_{31} 신규취약점이 발견되는 것을 가정 할 수 있다.

이 검증을 통해, 공격을 시작하는 V_0노드에서 V_2가 존재하는 H2로 감염될 확률이 가장 높다는 것을 직관적으로 파악할 수 있다. 이러한 개념을 공격그래프상 노드에 존재하는 소프트웨어 취약점 발견율에 적용할 수 있다. Fig.12를 보면, V_0에서 V_2로 이어지는 간선이 더 높은 가중치 점수를 가지는 것을 파악할 수 있다. 이를 공격그래프상 노드를 더 두껍게 표시하였으며, 이를 통해 V_0가 존재하는 노드에서 V_2가 존재하는 H2로의 전이확률이 H1,H3으로 전이확률보다 더 크다는 것을 알 수 있다.

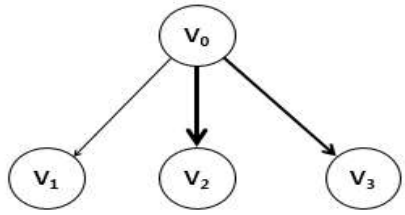

Fig. 12. Case2-Attack Graph combined with VDM

5. 결론

일반 기업의 IT환경에서는 지난 수 십년 동안 취약점이 발견되면 이를 즉각 패치 하는 것이 당연한 것으로 생각하고 있었다. 하지만 계측제어시스템 등이 포함된 핵심기반시설에서는, 취약점 발견과 이에 따른 패치가 쉽지 않아 보안조치 및 강화에 큰 효과를 가져오지 못하였다. 현실에서는 계측제어시스템 상의 취약점을 패치하는 것은 불가능할 정도로 힘들다. 최근 도입되는 계측제어 시스템은 대부분 Windows등의 상용 운영체제가 설치되어 있으며, 따라서 패치를 했을 때 오류 발생이나 제어 어플리케이션의 안정성을 보장할 수 없어 대부분 계측제어시스템 운영사들은 패치를 주저하고 있는 것이 사실이다.

이런 계측제어시스템에 대한 취약점 패치를 효과적으로 수행하기 위해서는, 먼저 취약점을 패치해도 전체 시스템의 안정성에 문제가 없다는 것을 증명하기 위해 테스트베드 등에서 검증이 우선되어야 한다. 또한 매일 수 없이 발생하는 취약점들의 패치 우선 순위 및 패치를 미 시행시 발생하는 위험의 크기를 정량적으로 파악하기 위해 본 연구에서 제안

하는 확률론적 취약점 평가 방안을 사용한다.

확률론적 취약점 평가 모델링은, 먼저 취약점의 생명주기를 파악하고, 공격 그래프에서 침해 가능한 공격 경로에 존재하는 취약점들 간 시간의 경과에 따른 위험도 크기를 계산하는 것이다. 이러한 과정을 통해 정량적 취약점 영향도 분석을 한 이후, 취약점 패치의 우선순위 및 시간의 경과에 따른 계측제어시스템의 취약점 패치 미 시행으로 인한 잔재 위험의 크기를 분석 할 수 있다.

본 연구에서는 계측제어시스템의 상용운영체제에 존재하는 취약점을 대상으로 시간의 경과에 따른 확률론적으로 취약점의 위험도 크기를 평가하는 방안을 제안하였다. 기존 일반 IT시스템 환경에서 적용해 왔던, 시스템의 신규 취약점을 발견하고, 이에 따른 취약점에 대한 패치를 신속하게 적용하는 프로세스가 아닌, 계측제어시스템에 대한 실제 공격 경로 상에 존재하는 노드들에 존재하는 취약점 간의 관계를 분석한 후, 시간 경과에 따른 취약점의 감염 특징을 동적으로 계산하여 확률론적으로 모델링하였다. 이러한 모델링 기법을 이용하여, 공격 그래프를 기반으로 마르코프 모델을 접목한 확률론적 취약점 평가 기법을 제안하였다. 제안한 모델은 시간의 흐름에 따른 취약점의 위험도를 정량적으로 계산하며, 상용운영체제의 취약점 기준으로 광범위하게 사용하는 공통취약점 평가체계를 사용하여 복잡한 취약점의 세부 요소들을 일관적으로 본 모델에 반영하였다.

본 연구에서는 이러한 확률론적 취약점 평가 모델링에 대한 프레임워크 및 프로세스, 수도(pseudo)알고리즘을 제시하였다. 향후 제안한 모델링을 이용하여, 실제 계측제어시스템을 모사한 테스트베드에서 실험이 필요하며, 시간의 흐름에 따른 취약점 위험도 크기 및 공격 경로상 위험의 동적 분석 등이 필요하다.

실제 핵심기반시설등의 현장에서, 확률론적 취약점 평가 모델링을 통해 도출된 결과를 바탕으로, 즉각 취약점 패치가 어려운 계측제어시스템 내에서 취약점 패치의 우선 순위 및 패치를 미 시행시 감내 가능한 취약점 크기의 임계값 파악, 공격 그래프상의 경로에 대한 취약점 전파를 실증해 볼 필요가 있다.

REFERENCES

[1] S. Y. Oh. & J. K. Hong. (2018). Vulnerability Case Analysis of Wireless Moving Vehicle. *Journal of the Korea convergence society, 9(8)*, 41-46.
DOI : 10.15207/JKCS.2018.9.8.041

[2] J. K. Cho. (2019). Study on Improvement of Vulnerability Diagnosis Items for PC Security Enhancement. *Journal of Convergence for information Technology, 9(3)*, 1-7.
DOI : 10.22156/CS4SMB.2019.9.3.001

[3] Recommended Practice for Patch Management of Control Systems. (2008). *Department of Homeland Security.* (pp. 23-24).

[4] L. S. IS. (2018). *Digital I&C System Diagram.* LS IS Product.
http://www.lsis.com/ko/product/view/P01211

[5] Pubudu et al. (2018). Non-Homogeneous Stochastic Model for Cyber Security Predictions. *The Journal of Information Security.* (pp. 12-24).
DOI : 10.15207/JKCS.2018.9.8.041

[6] Karen Scarfone. (2009). An analysis of CVSS version 2 vulnerability scoring. *ESEM '09 Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement.* (pp. 516-525).
DOI : 10.1109/ESEM.2009.5314220

[7] S. M. Rajasooriya & C. P. Tsokos. (2017). Cybersecurity: Nonlinear Stochastic models for Predicting the Exploitability. *The Journal of information Security. (*pp. 125-140).
DOI : 10.4236/jis.2017.82009

[8] P. Ammann. (2002). Scalable, graph-based network vulnerability analysis. *Proceedings of the 9th ACM conference on Computer and communications security.* (pp. 217-224).
DOI : 10.1145/586110.586140

[9] S. Jah. (2002). Two formal analyses of attack graphs. *The Proceedings 15th IEEE Computer Security Foundations Workshop.*
DOI : 10.1109/CSFW.2002.1021806

[10] S. Abraham. & S. Nair. (2014). Cyber Security Analytics: A Stochastic Model for Security Quantification Using Absorbing Markov Chains. *Journal of Communications, 9(12),* 899-907.
DOI : 10.12720/jcm.9.12.899-907

[11] A. Reibman & K. Trivedi. (1998). Numerical transient analysis of markov models. *Computer & Operations*

Research, 15(1), 19-36.
DOI : 10.1016/0305-0548(88)90026-3

[12] B. A. Craig. (2002). Estimation of the transition matrix of a discrete time Markov chain. *Health Economics, 11(1),* 33-42.
DOI : 10.1002/hec.654

[13] S. Swapna. (2004). Analysis of Software Fault Removal Policies Using a Non-Homogeneous Continuous Time Markov Chain. *Software Quality Journal, 12(3).* (pp. 211-230).
DOI : 10.1023/B:SQJO.0000034709.63615.8b

[14] A. Andan & S. Munmad. (2005). Verifying continuous time Markov chains. *International Conference on Computer Aided Verification.* (pp. 269-276).
DOI : 10.1007/3-540-61474-5_75

[15] G. Laurent. (2011). Vulnerability Discrimination Using CVSS Framework. *2011 4th IFIP International Conference on New Technologies, Mobility and Security.*
DOI : 10.1109/NTMS.2011.5720656

[16] S. Roger. (1989). Markov and Markov reward model transient analysis: An overview of numerical approaches. *European journal of Operation Research, 40(2).* 257-267.
DOI : 10.1016/0377-2217(89)90335-4

[17] N. Skku. (2015). Exploitability analysis using predictive cyber security framework. *2015 IEEE 2nd International Conference on Cybernetics.*
DOI : 10.1109/CYBConf.2015.7175953

[18] J. Y. Kim. (2007). Vulnerability Discovery in Multi version software systems. *10th IEEE High Assurance Systems Engineering Symposium..*
DOI : 10.1109/HASE.2007.55

엄익채(Ieck-Chae Euom) **[정회원]**

· 2003년 8월 : 전남대학교 컴퓨터 정보학과(이학사)
· 2015년 2월 : 한국과학기술원 전산학과(공학석사)
· 2019년 2월 : 전남대학교 정보보안협동과정(이학박사)

· 2007년 9월 ~ 현재 : 한전KDN(주) 보안컨설팅팀 재직중
· 관심분야 : 산업제어시스템 보안, 소프트웨어 개발 보안
· E-Mail : icelaken@gmail.com