JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 3, pp. 65-72, March 2020
https://doi.org/10.9708/jksci.2020.25.03.065

Design and Implementation of Cyber Attack Simulator based on Attack Techniques Modeling

Yong Goo Kang*, Jeong Do Yoo*, Eunji Park*, Dong Hwa Kim**, Huy Kang Kim*

*Student, Graduate School of Information Security, Korea University, Seoul, Korea
*Student, Graduate School of Information Security, Korea University, Seoul, Korea
*Student, Graduate School of Information Security, Korea University, Seoul, Korea
**Senior Researcher, The 2nd R&D Institute, Agency for Defense Development, Seoul, Korea
*Professor, Graduate School of Information Security, Korea University, Seoul, Korea

[Abstract]

With the development of information technology and the growth of the scale of system and network, cyber threats and crimes continue to increase. To cope with these threats, cybersecurity training based on actual attacks and defenses is required. However, cybersecurity training requires expert analysis and attack performance, which is inefficient in terms of cost and time. In this paper, we propose a cyber attack simulator that automatically executes attack techniques. This simulator generates attack scenarios by combining attack techniques modeled to be implemented and executes the attack by sequentially executing the derived scenarios. In order to verify the effectiveness of the proposed attack simulator, we experimented by setting an example attack goal and scenarios in a real environment. The attack simulator successfully performed five attack techniques to gain administrator privileges.

▸ **Key words**: Cybersecurity, Security Training, Modeling, Automation, Simulation

[요 약]

정보 기술의 발달과 시스템 및 네트워크의 규모가 증가함에 따라 사이버 위협 및 범죄가 꾸준히 증가하고 있다. 이러한 위협에 대응하기 위해서 실질적인 공격과 방어 기반의 사이버 보안 훈련이 필요하다. 그러나 사이버 보안 훈련은 전문가의 분석과 공격 수행능력을 요구하므로, 비용 및 시간적인 측면에서 비효율적이다. 본 논문에서는 공격 기법들을 자동으로 수행하는 사이버 공격 시뮬레이터를 제안한다. 이 시뮬레이터는 구현 가능하도록 모델링 된 공격 기법들을 조합하여 공격 시나리오를 도출하고, 도출된 시나리오들을 순차적으로 수행함으로써 공격을 수행한다. 제안하는 공격 시뮬레이터의 유효성을 검증하기 위해 실제 환경에서 예시 공격 목표와 시나리오를 설정하여 실험하였다. 이 공격 시뮬레이터는 5 가지 공격 기법을 자동으로 수행하여 관리자 권한을 획득하는 공격에 성공하였다.

▸ **주제어**: 사이버보안, 보안 훈련, 모델링, 자동화, 시뮬레이션

• First Author: Yong Goo Kang, Corresponding Author: Huy Kang Kim
*Yong Goo Kang (yonggoo@korea.ac.kr), Graduate School of Information Security, Korea University
*Jeong Do You (opteryx25104@korea.ac.kr), Graduate School of Information Security, Korea University
*Eunji Park (epark911@korea.ac.kr), Graduate School of Information Security, Korea University
**Dong Hwa Kim (dhkim@add.re.kr), The 2nd R&D Institute, Agency for Defense Development
*Huy Kang Kim (cenda@korea.ac.kr), Graduate School of Information Security, Korea University
• Received: 2020. 02. 28, Revised: 2020. 03. 04, Accepted: 2020. 03. 12.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

HACKMAGEDDON의 보고서와 CVE-detail의 통계에 따르면 사이버 보안에 대한 월별 사건의 수가 지속적으로 증가하고 있다[1]. 보안 전문 업체인 Risk Based Security의 보고서에 따르면 2018년 상반기 동안 새롭게 발견된 사이버 취약점의 수는 10,644개로 매년 1만개 이상에 달하는 취약점이 새롭게 발견되고 있다[2]. 이런 현상은 기업의 이익 뿐 아니라 국가의 안보를 지키기 위하여 사이버 위협에 대한 국가 차원의 효율적인 대비가 필요하다는 것을 보여준다. 따라서 사이버 보안 인력의 역량을 강화하기 위해 사이버 공격과 방어에 대한 교육 및 훈련의 필요성이 증가하고 있다. 하지만 실제 운용되는 환경을 대상으로 이를 수행하는 것은 큰 위험 부담 및 제약이 발생하며, 현실적으로 불가능하다. 또한 전문가의 분석과 전문적인 공격 역량이 요구되어 인력, 시간, 그리고 비용 측면에서 비효율적이다.

이러한 문제점을 해결할 수 있는 것이 시뮬레이션 기반의 사이버전 훈련장이다. 국내 뿐 아니라 미국 등 선진국에서는 사이버전 훈련장을 자체 개발하여 훈련 및 시험을 위한 테스트베드를 지원한다. 다만, 훈련 시스템의 세부 사항을 파악하는 것이 쉽지 않고, 목적에 따른 다양한 환경 및 시나리오를 자율적으로 설정하여 훈련을 수행하기는 어려운 실정이다. 따라서 훈련에 필요한 공격 기법을 자율적으로 추가하고, 다양한 공격 시나리오를 만들어 수행할 수 있는 사이버 공격 시뮬레이터의 개발이 요구되고 있다.

본 논문에서는 사이버 공격 기법을 모델링하여 다양한 공격 시나리오를 도출하고 자동으로 수행할 수 있는 공격 시뮬레이터를 설계하고, 설계된 시뮬레이터가 실제 환경에서 잘 동작함을 보여 유효성을 검증하여, 향후 사이버 보안 훈련을 위한 공격자 역할을 대체할 수 있는 가능성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 국내외 사이버 훈련 시스템의 현황과 관련 연구를 알아본다. 3장에서는 제안하는 공격 기법 모델링과 사이버 공격 시뮬레이터의 구성 및 알고리즘을 기술한다. 4장에서는 실제 테스트 환경을 구축하여 제안하는 공격 시뮬레이터가 정상적으로 공격 목표를 달성함을 보인다. 5장에서는 본 연구의 한계점을 분석하고, 6장에서는 결론과 향후 나아가야할 방향을 다룬다.

II. Preliminaries

1. Related works

이스라엘 사이버짐은 2013년에 설립된 세계 최초의 사이버 보안 에뮬레이션 훈련을 제공하는 훈련 센터이다[3]. 훈련은 공격을 담당하는 레드팀, 방어를 담당하는 블루팀, 훈련에 대한 모니터링을 수행하는 화이트팀으로 구성되며 훈련생은 블루팀의 역할을 수행한다. 훈련은 고객의 사이버 위협 모델을 통해 사용조직 내에서 사용 중인 정책과 기술, 그리고 도구들을 고려하여 조직원들이 사이버 공격에 대응할 수 있는 역량을 확보하는 것을 목표로 한다.

NATO의 locked Shield는 기술 전문가, 군 관련자, NATO 가입국 의사결정권자들을 위하여 사이버 방어 훈련을 조직하는 CCDCOE(Cooperative Cyber Defense Centre of Excellence)에 의하여 2010년부터 매년 개최되는 훈련이다[4]. 훈련 시나리오는 취첨단 기술 및 해킹 기술들을 통해 실제로 발생할 수 있는 수준으로 복잡하게 구성된다.

Caldera는 1958년에 설립된 미국의 비영리 단체인 MITRE에서 연구하고 개발한 Windows enterprise network에 대한 자동화된 공격을 수행할 수 있는 오픈소스 공격 플랫폼이다[5]. 공격자가 설정한 공격 목표, 공격 기법 등을 고려하여 자동으로 공격을 수행한다. 공격 과정에 사용하는 공격 기법 및 커맨드는 MITRE ATT&CK의 자료를 기반으로 한다[6]. 다만, 공격 대상의 네트워크에서 최소 하나의 호스트는 이미 공격에 성공했다고 가정한다.

유럽에서는 2010년부터 2년 주기로 사이버 훈련을 시행하였으며, ENISA(European Network and Information Security Agency)가 유럽 연합 집행 위원회로부터 권한을 위임받아 훈련을 실시하고 있다[7]. 아태지역 침해사고 대응팀 협의회(APCERT, Asia Pacific Computer Emergency Response Team)는 2011년부터 매년 1년 씩 사이버 대응훈련을 실시하고 있으며, 2019년 6월 기준으로 한국(KISA), 일본, 호주 등 21개 국가로 구성되어 있다[8].

Z. C. Schreuders et al.[9]에서는 다양한 use cases를 이용해 임의의 공격 시나리오에 대한 가상머신을 만드는 방법을 제공한다. 취약한 시스템을 공격하거나 방어하기 위한 환경을 효율적으로 만들 수 있다. J. Mirkovic et al.[10]에서는 사이버 보안 실험을 행위, 구조, 제약사항으로 나누어 인코딩하는 방법을 제안하여 기존의 특정 환경에만 한정되어 있는 테스트베드 기반 실험의 한계를 완화시킨다. S. Wi et al.[11]에서는 git을 기반으로 간단하고 효율적인 CTF를 제안하여 보안 대회를 운영하는 데 드는 시간과 인력을 줄이고자 하였다. E. Trickel et al.[12]에서는 보안 교육을 효율적으로 진행하고자 하는 관점에서 ATCTF(Attack/Defense CTF)를 손쉽게 개최할 수 있도록 도구를 개발하고 오픈소스로 공개하였다.

III. The Proposed Scheme

본 연구에서는 사이버 공격 기법을 수행하기 위해 필요한 개념을 모델링하고, 이를 기반으로 공격 목표와 공격 대상의 상태에 따라 가능한 공격 기법들의 집합들을 자동으로 수행할 수 있는 사이버 공격 시뮬레이터를 설계하고 구현한다. 본 장에서는 공격 기법 모델링, 공격 시뮬레이터 설계와 알고리즘을 기술한다.

1. Attack Techniques Modeling

특정 공격 목적을 달성하기 위하여 여러 공격 기법이 연결될 필요가 있다. 또한 연결된 공격 기법들에 대하여 각 기법이 수행될 때마다 공격의 대상이 어떻게 변해 가는지를 기록해야 한다. 이를 위하여 본 연구에서는 공격 기법 모델링을 위해 특정 공격 기법을 수행할 수 있는 사전 조건(precondition), 공격 기법을 수행하고 난 뒤의 사후조건(postcondition), 그리고 시뮬레이터가 공격 대상의 현재 상태를 기록하는 정보(state)를 정의하였다.

1.1 State

상태(state)는 키(key)와 값(value)의 쌍으로 된 요소들의 집합으로써, 공격자가 공격 대상에 대해 어떤 정보를 알고 있는지를 나타낸다. 각 요소는 공격 대상에 대한 무엇(key)이 어떻게(value) 되어 있다고 해석한다. 표 1은 상태에 포함될 수 있는 키 목록을 나타낸다. 예를 들어, 하나의 요소가 *{OS, linux}* 라면, 공격 대상의 운영체제가 Linux 라고 해석한다.

주소(address)는 대상의 네트워크 주소를 의미한다. 프로토콜에 따라 형태가 달라질 수 있으며 프로토콜이 IP인 경우 *X.X.X.X* 형태를 가진다. 프로토콜(protocol)은 대상에 접근하기 위한 통신 프로토콜을 의미한다. 본 연구에서는 IP만 고려하였으나 향후 다양한 프로토콜이 추가될 수 있다. 운영체제(os)는 대상의 운영체제를 의미한다. 공격 시뮬레이터가 공격 기법의 집합을 기반으로 동작하기 때문에 운영체제를 알 필요가 있으며, 이 정보로 사용할 수 있는 취약점을 유추할 수 있다. 포트(ports)는 대상에 접근할 수 있는 포트를 의미하며, IP 프로토콜인 경우에 사용된다. 포트 값이 2개 이상인 경우도 가능하다. 쉘(shell)은 대상에 접근하기 위한 쉘의 상태를 의미하며, 다음의 3가지 값을 가질 수 있다. *TEMPORAL*은 쉘을 일시적으로 획득한 상태, *DOWNLOADED*는 영구적인 쉘 획득을 위해 별도의 프로그램(RAT, Remote Manipulator System)이 대상에 전송된 상태, *PERMANENT*는 영구적은 쉘을 획득한 상태를 의미한다. 권한(privilege)은 대상으로부터 확보한 권한을 의미하며, 시스템 권한과 사용자 권한으로 구분된다. 바이너리(binaries)는 대상에서 사용 가능한 프로그램 목록을 의미하며, 공격 기법을 선택하는데 중요하게 사용된다. 경로(cwd)는 대상에 접근한 쉘의 현재 경로를 의미한다. 수도(misudo)는 sudo 설정에서 악용 가능한 요소가 무엇인지를 의미한다.

Table 1. Key list and description of state model

Key	Description
address	Address of the target
protocol	Protocol for access to the target
os	OS of the target
ports	Ports for access to the target
shell	State of shell to the target
privilege	Acquired privilege for the target
binaries	Available program list on target
cwd	Current working directory of shell
misudo	Exploitable points on sudo config

1.2 Precondition

사전조건(precondition)은 공격 기법을 수행하는데 필요한 조건을 정의한다. 공격 대상의 상태와 구현된 각각의 공격 기법의 사전조건을 조사하여 조건을 만족하면 해당 공격 기법을 수행할 수 있다. 예를 들어, *cron*을 설치하는 공격 기법은 현재 상태에 *{shell, DOWNLOADED}, {binaries, [cron, ..]}, {privilege, 0}* 과 같은 값이 설정된 상태여야 수행할 수 있다.

1.3 Postcondition

사후조건(postcondition)은 공격 기법이 성공적으로 수행된 이후 만족할 조건들을 정의한다. 단, 사후조건은 실제 공격을 수행하기 전 공격 시나리오를 도출하는 과정에서 공격이 성공적으로 수행되었다고 가정하기 위해 사용된다. 따라서 어떤 키에 대한 정보를 얻을 것인지는 사전에 정의가 가능하지만, 실제 어떤 값인지는 사전에 알 수 없는 경우가 대부분이다. 이 경우 *UNKNOWN* 값을 사용한다. 예를 들어, 시스템 조사 공격 기법을 수행하면 *{privilege, UNKNOWN}, {binaries, UNKNOWN}, {cwd, UNKNOWN}* 요소들이 상태에 추가된다. 이렇게 요소들을 추가하면 다음 공격 기법의 사전조건을 만족시켜 이어지는 공격에 대한 도출을 수행할 수 있다.

2. Simulator Design

사이버 공격 시뮬레이터의 구성요소는 그림 1과 같다.

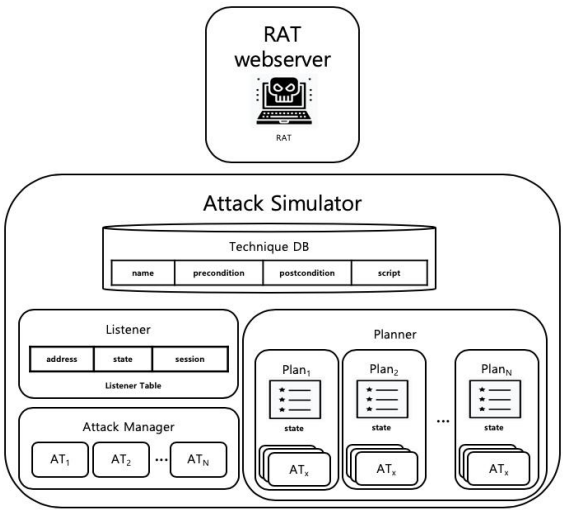

Fig. 1. Components of Attack Simulator

RAT 웹서버(webserver)는 원격지에서 RAT를 다운 받을 수 있도록 만들어진 웹 서버이다. 기법 저장소(Technique DB)는 프로그램화 된 공격 기법들을 보유하고 있는 저장소이다. 각 공격 기법은 명칭(name), 사전조건, 사후조건, 스크립트(script) 정보를 보유한다. 스크립트는 공격 수행을 위해 실행되어야 할 별도의 파일에 대한 경로를 의미한다. 리스너(Listener)는 공격 시뮬레이터가 보유하고 있는 공격 대상의 쉘을 관리하기 위하여 존재한다. 대상 시스템에 대한 리버스 쉘(reverse shell) 공격은 리스너가 열고 있는 포트로 접속하는 방식으로 이루어지며, 이렇게 연결된 세션은 리스너에 의해 관리된다. 공격 관리자(Attack Manager)는 공격 시뮬레이터가 동작할 때 기법 저장소에 있는 모든 데이터를 AT라는 클래스로 인스턴스를 생성시키고 관리한다. 플래너(Planner)는 공격 목표를 달성할 수 있는 모든 시나리오를 도출하여 플랜으로 저장하고 순차적으로 시도할 수 있도록 관리한다.

3. Algorithms

사이버 공격 시뮬레이터의 알고리즘은 전체 과정, 시나리오 도출 과정, 시나리오 수행 과정으로 구분할 수 있다.

3.1 Overall Process

전체 과정은 그림 2와 같이 시나리오 도출 과정과 시나리오 수행 과정을 포함한 시뮬레이터의 전체적은 동작 과정을 나타낸다. 알고리즘의 결과는 공격의 성공 여부를 나타낸다. 도출한 공격 시나리오 중 최소 하나가 공격 목표를 달성하는 경우 공격이 성공한 것으로 간주하며, 모든 시나리오가 목표를 달성하지 못하는 경우 실패한 것으로 간주한다. *initialization*은 공격 기법들을 인스턴스로 생성하는 등의 시뮬레이터의 초기화 과정을 수행한다.

Algorithm 1: The overall process

```
initialization();
scenarios ← generate_scenarios(...);
foreach scenario ∈ scenarios do
    result ← perform_scenario(scenario);
    if result is true then
        print("Attack Succeeded");
        return;
    end
end
print("Attack Failed");
```

Fig. 2. Algorithm of the overall process

3.2 Generate Scenarios

시나리오 도출 과정은 그림 3과 같이 공격 목표를 달성할 수 있도록 보유하고 있는 공격 기법들을 서로 연결하는 작업을 수행한다. 공격 기법의 종류와 순서가 달라지는 경우 서로 다른 시나리오로 간주되어 여러 개의 공격 시나리오가 생성될 수 있다. 입력으로 들어오는 state는 시나리오를 생성하기 위해 기초가 되는 상태, attacks는 사용 가능한 모든 공격 기법들, *sequence*는 현재까지 도출된 공격 기법의 순서를 의미한다. *get_available_attacks* 를 통해 주어진 상태와 공격 기법들의 사전조건을 조사하여 수행 가능한 공격 기법들을 추출한다. 추출된 각 공격 기법별로 수행하였다고 가정하고 사후조건을 상태에 반영하여 공격 시나리오 도출 함수를 재귀적으로 호출한다. 이때, *attacks*에 현재 선택한 공격을 제외하여 중복으로 선택되는 것을 방지한다. 선택한 공격은 *sequence*에 추가하여 재귀적으로 호출될 때 축적될 수 있도록 한다. 더 이상 선택할 공격 기법이 없을 경우 재귀적 호출을 종료하며, 그 순간의 상태가 공격 목표를 달성하면 생성된 *sequence*를 *scenarios*에 추가한다. 최종적으로 공격 목표를 달성할 수 있는 *sequence*들이 공격 시나리오로 도출된다.

Algorithm 2: generate_scenarios

```
Input: state
Input: attacks (all available attack techniques)
Input: sequence (cumulative attack technique sequence)
Output: scenarios (available attack scenarios)
scenarios ← {};
if is_goal_achieved(state) then
    new_scenario ← make_scenario(sequence);
    scenarios ← scenarios + new_scenario;
    return scenarios;
end
next_attacks ← get_available_attacks(state, attacks);
foreach next_attack ∈ next_attacks do
    copied_state ← state;
    copied_attacks ← attacks;
    copied_sequence ← sequence;
    copied_state ← update_postcondition(copied_state,
      next_attack);
    copied_attacks ← copied_attacks - next_attack;
    copied_sequence ← copied_sequence + next_attack;
    result_scenario ← generate_scenarios(copied_state,
      copied_attacks, copied_sequence);
    scenarios ← scenarios + result_scenario;
end
```

Fig. 3. Algorithm of the scenario generation

3.3 Perform Scenarios

시나리오 수행 과정은 그림 4와 같이 도출한 공격 시나리오들을 순차적으로 수행한다. *perform_attack*을 통해 각 시나리오가 보유한 공격 기법들을 실제 환경에서 순차적으로 실행하고, 각 기법의 사후조건에 따라 상태를 업데이트 한다. 만일 현재 수행할 공격 기법의 사전조건이 상태에 만족하지 못하면 해당 시나리오는 실패로 판단하고 종료한다. 시나리오의 모든 공격 기법을 수행하고, 최종 상태가 공격 목표를 만족하면 공격을 성공으로 판단하고 종료한다. 모든 공격 기법을 수행한 뒤에 공격 목표를 만족하지 못하면 실패로 판단한다. 도출한 모든 시나리오가 *perform_scenario*를 통해 공격 목표를 만족하지 못하면 최종적으로 공격을 실패했다고 판단한다.

Algorithm 3: perform_scenario

Input: $scenario$ (attack sequence that can satisfy the goal)
$state \leftarrow \{\}$;
foreach $attack \in scenario$ **do**
  if ***is_goal_achieved***$(state)$ **then**
    return true;
  end
  if ***not is_precondition_satisfied***$(state, attack)$ **then**
    return false;
  end
  $state \leftarrow$ ***perform_attack***$(state, attack)$;
end
return false;

Fig. 4. Algorithm of scenario execution

IV. Implementation

본 연구에서는 제안하는 사이버 공격 시뮬레이터가 정상적으로 동작하는지 검증하기 위하여 Virtual Box(6.0.14) 와 Ubuntu(18.04.3) 환경에서 실제로 구현하여 테스트하였다. 테스트를 위해 특정 공격 목표를 설정하고, 해당 목표를 달성하기 위한 5가지 공격 기법들의 집합을 예시 시나리오로 설정하였다.

1. Attack Goal

본 연구에서는 표 2와 같이 공격 대상 호스트에 대한 영구적인 쉘을 관리자 권한으로 획득하는 것을 공격 목표로 설정하였다. 예시 시나리오의 원활한 동작을 위해 공격 대상 호스트의 환경이 표 3과 같다고 가정하여 사전에 설정하였다. 즉, 공격 대상 호스트에는 IP 프로토콜을 통해 접근이 가능하고, IP 주소는 표와 같으며, 운영체제는 Linux이고, 마지막으로 *wget*, *cron*과 같은 프로그램이 존재한다.

Table 2. Attack goal for a sample scenario

Key	Value
address	163.152.127.210
privilege	0 (root)
shell	PERMANENT

Table 3. Environment of target host

Key	Value
protocol	ip
address	163.152.127.210
os	linux
binaries	wget, cron, etc.

2. Attack Techniques

예시 시나리오의 공격 목표를 달성하기 위해 구현한 공격 기법들은 표 4와 같다.

Table 4. Attack techniques for the sample scenario

ID	Technique Name
AT1	Nmap for searching open ports
AT2	CVE-2016-3714 for RCE (unix)
AT3	System Investigation (unix)
AT4	Download (wget)
AT5	Install (cron)

AT1은 *nmap*을 통해 공격 대상 호스트에 열려있는 포트를 확인하는 기법이다. AT2는 Unix 계열의 호스트에 대해 CVE-2016-3714 취약점을 통해 원격 명령을 수행하도록 하는 기법이며, 임시적인 쉘을 획득할 수 있다. AT3은 Unix 계열의 호스트에 대한 시스템의 환경을 조사하는 기술이다. AT4는 *wget*을 이용하여 공격자 호스트의 RAT를 공격 대상 호스트에 내려 받는 기법이다. AT5는 *cron*을 이용하여 공격자 호스트에서 RAT를 영구적으로 동작시키는 기법이며, 영구적인 쉘을 획득할 수 있다. 본 연구에서는 이러한 공격 기법들을 순차적으로 수행하는 예시 시나리오를 통해 공격 목표를 달성할 수 있음을 보인다.

3. Simulation Result

그림 5는 공격 시뮬레이터를 동작시킨 후 가능한 공격 시나리오들을 도출하는 과정을 나타낸다.

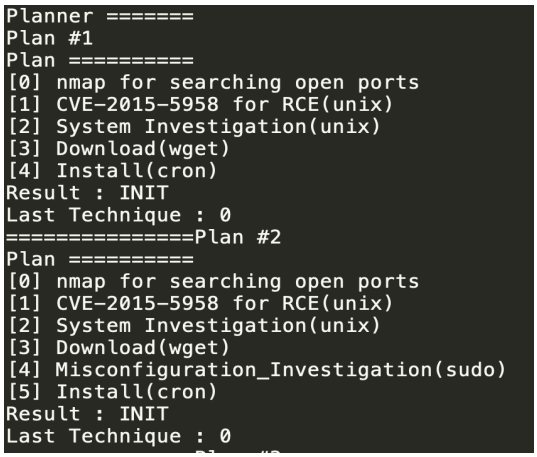

Fig. 5. Example of generated scenarios

그림 6은 예시 시나리오의 첫 번째 공격 단계인 AT1 공격 기법을 수행한 결과 화면이다. 처음에는 초기에 주어진 최소 정보인 프로토콜과 주소만 가지고 있음을 볼 수 있다. 공격 기법의 사전조건의 운영체제, 포트가 *UNKNOWN* 이지만, 실제로 공격을 수행한 이후에는 운영체제가 Linux이고, 포트가 다수 조회된 것을 확인할 수 있다.

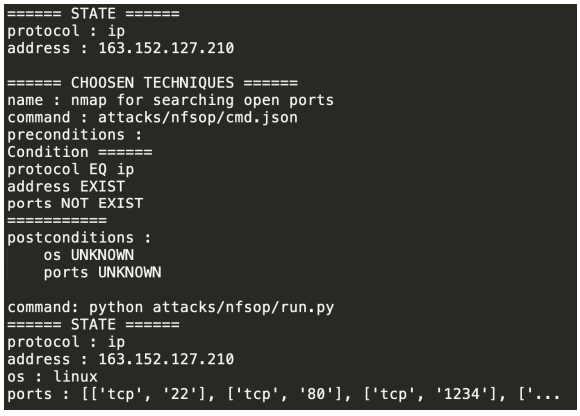

Fig. 6. Result of performing attack technique AT1

이와 같은 원리로 도출된 시나리오를 순차적으로 수행한다. 그림 7은 최종적으로 공격 시뮬레이터가 목표를 달성하여 공격이 성공하였음을 나타내며, 실제로 표와 같은 공격 기법들이 순차적으로 수행되었음을 확인하였다.

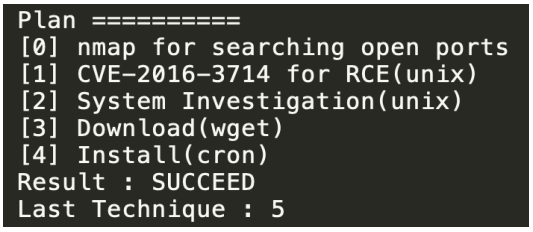

Fig. 7. Final result of performing attack simulator

그림 8은 공격이 성공한 이후 공격자 호스트에서 *nc* 명령을 통해 공격 대상 호스트의 쉘을 획득한 장면이다. 호스트의 IP 주소가 공격 대상의 IP 주소와 일치하고, 현재 접속 계정의 ID 가 0 이므로 관리자 권한임을 확인하였다.

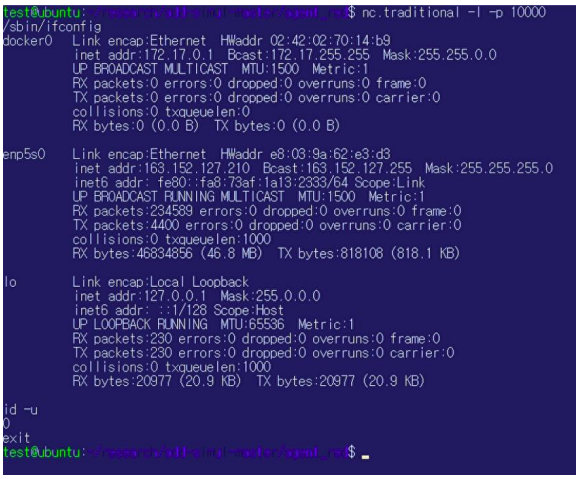

Fig. 8. Shell connection after the attack

이와 같이 공격 기법과 상태를 모델링함으로써 사이버 공격 시뮬레이터가 공격 기법들을 자동적으로 수행하고, 공격 목표를 정상적으로 달성하는 것을 확인하였다.

V. Discussion

공격 기법 범위 확장: 사이버 공격 기법의 종류와 수는 파악하기 어려울 정도로 많이 존재한다. 실험을 통해 5가지 공격 기법을 사용함으로써 모델링과 공격 시뮬레이터의 구현 가능성을 검증하였으나, 실제로 훈련 또는 교육 목적으로 시뮬레이터를 사용하기 위해서는 더 많은 공격 기법들이 구현되어야 한다.

모델링 확장: 광범위한 공격 기법을 적용하고자 할 때, 본 연구에서 정의한 상태 모델링 기반으로는 설명하기 어려운 공격 기법이 존재할 수 있다. 공격 기법의 확장에 따라 필요한 상태 요소를 수정해야 하는데, 단순하게 키를 정의하여 추가하거나 충분히 고민하지 않으면 추후 비슷한 내용이 중복되면서 이를 수정하기에 많은 비용이 소요될 수 있다. 따라서, 확장하고자 하는 공격 기법들을 최대한 조사하여 일정 수준 이상의 공격 시나리오들을 커버하는 만족스러운 수준이 되었을 때, 추가해야 할 상태 요소를 분석하고 반영할 필요가 있다.

멀티 홉 공격: 본 연구는 공격자와 공격 대상의 호스트가 직접적으로 연결되어 있는 단일 홉 네트워크 환경을 가정한다. 실제 공격 환경에서는 공격 대상이 단일 홉이 아닌 멀티 홉 거리에 존재하는 경우가 대부분이다. 따라서 접근 가능한 호스트를 순차적으로 공격하여 최종적으로 공격 목표에 도달하는 피버팅(pivoting) 기능 등을 이용한 공격 기법을 추가할 필요성이 있다. 멀티 홉 환경에서의 피버팅 기능을 추가하면 다양한 네트워크 토폴로지에서의 자동화된 공격이 가능하다. 이를 활용하면 스마트 홈 네트워크와 같이 서비스되고 있는 실제 환경을 가정하여 본 공격 시뮬레이터를 통해 취약점을 분석할 수 있을 것으로 기대된다.

VI. Conclusions

본 논문에서는 공격 수행에 소요되는 비용을 줄여 효율적으로 보안 훈련을 운영하기 위한 사이버 공격 시뮬레이터를 제안하였다. 사이버 공격 기법을 구현 가능하도록 모델링하였고, 시뮬레이터의 구성요소와 알고리즘을 설계하였다. 실제 환경에서 시뮬레이터가 구현된 공격 기법들을 자동으로 수행하고 목표를 달성하는 것을 보여줌으로써 실용가능성을 검증하였다. 검증에 사용한 공격 목표는 공격 대상 호스트의 쉘을 관리자 권한으로 영구적으로 획득하는 것이고, 5가지 공격 기법을 수행하여 위 목표를 달성하였다.

본 연구에서는 테스트를 위한 환경이 제한적이고 소수의 공격 기법만을 구현하였으나, 추가하고자 하는 공격 기법을 제안한 모델링에 맞춰 구현하여 기법 저장소에 추가하면 시뮬레이터를 통해 자동으로 수행할 수 있음을 확인하였다. 또한 공격이 추가됨에 따라 도출할 수 있는 공격 시나리오가 풍부해지며, 이는 다양한 공격 유형을 자동화하여 훈련에 사용할 수 있음을 의미한다.

향후 멀티 홉 네트워크 환경에서도 피버팅 기능을 활용하여 동작하도록 확장하고자 한다. 그리고 잘 알려진 공격 기법들이 멀티 홉 환경에서 자동으로 수행될 수 있도록 연구를 진행할 것이다. 최종적으로는 훈련 뿐 아니라 스마트 홈 네트워크 등을 대상으로 한 취약점 분석의 용도로도 사용할 수 있을 것으로 기대한다.

ACKNOWLEDGEMENT

This work was supported by the Agency for Defense Development under the contract UD190002ED.

REFERENCES

[1] P. Passeri, A year of cyber attacks, Available at https://www.hackmageddon.com/2019/01/15/2018-a-year-of-cyber-attacks.

[2] Vulnerability Quick View Report, Available at https://pages.riskbasedsecurity.com/2018-midyear-vulnerability-quickview-report.

[3] Locked Shields, Available at https://ccdcoe.org/exercises/locked-shields.

[4] CYBERGYM, Available at https://www.cybergym.com.

[5] CALDERA, Available at https://www.mitre.org/research/technology-transfer/open-source-software/caldera.

[6] MITRE ATT&CK, Available at https://attack.mitre.org.

[7] European Network and Information Security Agency, Available at https://www.enisa.europa.eu.

[8] Asia Pacific Computer Emergency Response Team, Available at https://www.apcert.org.

[9] Z. C. Schreuders, T. Shaw, M. Shan-A-Khuda, G. Ravichandran, J. Keighley, and M. Ordean, “Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events,” In 2017 USENIX Workshop on Advances in Security Education (ASE 17).

[10] J. Mirkovic, G. Bartlett, and J. Blythe, “DEW: Distributed Experiment Workflows,” In 11th USENIX Workshop on Cyber Security Experimentation and Test (CSET 18).

[11] S. Wi, J. Choi, and S. K. Cha, “Git-based CTF: A Simple and Effective Approach to Organizing In-Course Attack-and-Defense Security Competition,” In 2018 USENIX Workshop on Advances in Security Education (ASE 18).

[12] E. Trickel, F. Disperati, E. Gustafson, F. Kalantari, M. Mabey, N. Tiwari, Y. Safaei, A. Doupe, and G. Vigna, “Shell We Play A Game? CTF-as-a-service for Security Education,” In 2017 USENIX Workshop on Advances in Security Education (ASE 17).

Authors

Yong Goo Kang received his B.S and M.S degrees in Computer Science and Engineering from Hanyang University, Republic of Korea, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree in Graduate School of Information Security from Korea University, Republic of Korea. He has been working at Samsung Electronics since 2011 and is currently a member of Samsung Research's Security Team. His current research interests are in data-driven security using machine learning.

Jeong Do Yoo received the B.S. degree in Computer Science from Korea University, Republic of Korea, in 2018. He is currently pursuing the M.S. degree and Ph.D. degree in Graduate School of Information Security, Korea University. His current research interests are in artificial intelligence and network security.

Eunji Park received the B.S. degree in Computer Science from Washington State University, Pullman, WA, USA in 2018. She is currently pursuing the M.S. degree in Graduate School of Information Security, Korea University, Republic of Korea. She is interested in data-driven security and behavior analysis.

Dong Hwa Kim received the B.S., M.S. degrees in School of Electrical Engineering from Korea University, Korea, in 2004, 2007. He is currently a senior researcher in Agency for Defense Development, Seoul, Korea. He is interested in cyber security and cyber training system.

Huy Kang Kim received the B.S. degree in industrial management and the M.S. and Ph.D. degrees in industrial and systems engineering from KAIST in 1998, 2000, and 2009, respectively. He founded A3 Security Consulting, the first information security consulting company in South Korea in 1999. He was a Technical Director and the Head of the Information Security Department, NCSOFT, from 2004 to 2010. He is currently a Professor with the Graduate School of Information Security, Korea University. His research interests include solving security problems in online games based on the user behavior analysis and data mining.