Journal of the KIMST, Vol. 23, No. 3, pp. 267-276, 2020
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2020.23.3.267

Research Paper 정보 · 통신 부문

사이버전 훈련을 위한 상태 저장 트래픽 발생 Architecture 설계 및 구현

홍수연*,1) · 김광수1) · 김태규1)

1) LIG넥스원㈜ 지능형SW연구소 사이버전연구팀

An Architecture Design and Implementation of Stateful Traffic Generation for Cyber Warfare Training

Suyoun Hong*,1) · Kwangsoo Kim1) · Taekyu Kim1)

1) *Cyber-warfare Research Team in Intelligent SW Research Center, LIGNex1 Co., Ltd., Korea*

(Received 1 April 2020 / Revised 21 May 2020 / Accepted 25 May 2020)

Abstract

Threats targeting cyberspace are becoming more intelligent and increasing day by day. To cope with such cyber threats, it is essential to improve the coping ability of system security officers. In this paper, we propose a stateful traffic generator that provide network background traffic for cyber warfare training. The proposed architecture is designed for generating traffic similar to real system traffic, so the trainee can perform more realistic training

Key Words : Cyber Warfare Training(사이버전 훈련), Stateful Traffic Generator(상태 저장 트래픽 발생기)

1. 서 론

정보가 공유되고 유통되는 사이버 공간은 디지털 시대인 현대에서 중요도가 나날이 증가하고 있다. 사이버 공간에 위협이 발생했을 경우, 시스템의 가용성, 무결성, 기밀성을 지키기 위해서는 상황의 신속하고 명료한 상황 파악과 공유가 선결되어야 한다. 이러한 능력을 향상시키기 위해서는 다양한 사이버 위협 시도를 적용한 실전적 훈련 방식이 필요하나 실제 운영되고 있는 시스템을 대상으로 이를 수행하기 위해서는 훈련을 위한 실제 사이버 위협이 진행되어야 하므로 많은 위험이 따른다. 이러한 훈련을 수행하기 위한 방안으로서 실제 구축 시스템과 유사한 형태의 훈련용 가상환경을 구축하여 사이버 훈련 수행 및 시스템의 공격 대응 능력을 검증하는 형태를 취한다[1]. 이러한 훈련용 가상환경을 의미하는 용어로 사이버레인지(Cyber Range)를 사용하며 미국 DAR PA(Defense Advanced Research Project Agency)가 개발한 국가 사이버전 시험장(NCR, National Cyber Range)[2]와 이스라엘에서 제공하는 사이버 보안 에뮬레이션 훈련을 위한 사이버짐[3]이 대표적 예이다.

훈련용 가상환경에서 실상황을 대비한 훈련을 하기 위해서는 실제 시스템과 유사한 형태의 환경을 구축

* Corresponding author, E-mail: suyoun.hong@lignex1.com

할 필요가 있다. 실제 시스템은 정적 요소인 네트워크 토폴로지(네트워크 토폴로지에 포함된 물리 장비)와 동적 요소인 유통 트래픽으로 구성되어 있으며 일반적인 사이버레인지에서는 네트워크 토폴로지를 유사하게 제공하는 것에 초점을 맞추고 있다. 그러나 훈련자가 외부에서 침투하는 사이버 위협을 식별하고 대처하는 훈련을 하고자 할 경우, 훈련 대상인 사이버 위협의 전달이 감춰질 수 있는 배경 트래픽이 존재하지 않는다면 훈련자가 위협 트래픽을 식별하기 위한 별도의 작업을 수행할 필요가 없어 훈련이 기대하는 효과를 발휘하지 못할 수 있다.

본 논문에서는 실제와 유사한 배경 트래픽을 사이버 훈련 시스템에 공급하기 위한 상태 저장 트래픽 발생 Architecture의 설계와 구현 결과를 제시하고자 한다.

본 논문의 2장은 기존 트래픽 발생기 하드웨어, 소프트웨어에 대한 분석과 사이버 훈련 상황에 실제와 유사한 환경을 주기 위한 기존 연구에 대해 설명한다. 3장에서는 상태 저장 트래픽 발생 Architecture에서 필요한 트래픽 데이터셋, 훈련 환경에서 동작하며 트래픽을 공급하는 트래픽 에이전트, 에이전트를 제어하기 위한 모니터링 서버의 설계 개념과 소프트웨어 구조를 설명한다. 그리고 4장에서는 상태 저장 트래픽 발생 Architecture의 구현 및 시험 결과에 대해 설명하고 5장에서는 결론을 요약하였다.

2. 기존 연구 분석

본 장에서는 서론에서 간략히 설명한 기존 사이버 훈련 시스템 및 트래픽 발생기 개발 사례를 분석한다.

2.1 기존 사이버 훈련 시스템

발생 가능한 사이버 위협에 대한 대처 능력을 강화하는 실전적 훈련을 위한 사이버 훈련 시스템은 실제의 사이버 환경을 가상화 기술을 이용하여 동적으로 생성한 유사한 가상운용환경을 지원한다. 훈련은 공격을 담당하는 레드팀, 방어를 담당하는 블루팀, 훈련에 대한 모니터링을 수행하는 화이트팀으로 구성되어 운용되며 시스템 구축 예로는 이스라엘의 사이버짐[3]과 미국 DARPA의 국가 사이버전 시험장(NCR, National Cyber Range)[2] 등이, 국내 예로는 KISA의 시큐리티짐(Security-Gym)이 2017년 11월부터 공식 운영을 시작한 바 있다. 또한 이러한 사이버전 훈련 시스템에서 블루팀이 훈련자로 구성될 때 레드팀 역할을 맡아 시나리오에 의해 설정된 공격을 자동적으로 수행하는 에이전트에 대한 개발이 진행된 바 있다[4].

2.2 기존 트래픽 발생기

트래픽 발생기는 일반적으로 시스템의 네트워크 부하 시험과 사이버 위협 대처 능력을 보기 위한 목적으로 개발되며 소프트웨어, 하드웨어로 구현된다. 이 중 사이버 위협 트래픽을 제외한 L4(OSI(Open System Interconnection) 7 계층 구조 중 Transport Layer) - L7 (OSI 7계층 구조 중 Application Layer) 단의 트래픽 발생을 위한 도구에는 T-Rex, Pktgen, MoonGen, Ostinato 등이 존재한다[5]. T-Rex는 CISCO에서 개발한 DPDK (Data Plan Development Kit: 고속 패킷 처리를 위한 데이터 플레인 라이브러리와 네트워크 인터페이스 컨트롤러 드라이버 집합) 기반의 상태 저장 및 상태 비저장 트래픽 생성기로서 실제 트래픽 템플릿을 재생하거나 L4-L7 트래픽을 생성할 수 있다. 그러나 실행을 위한 별도의 하드웨어가 필요하다. pktgen은 리눅스 운영체제에서 제공하는 네트워크 부하 시험용 packet sender로서 고속으로 패킷을 생성하고 유통 통계 정보를 얻을 수 있는 특징을 가진다. MoonGen은 pktgen와 같이 네트워크 부하 시험용 packet sender로서 고속 패킷 생성, 유통 결과 통계 정보 제공의 기능을 가지며 Intel 계열의 NIC에 의존적인 특성을 가진다. Ostinato는 네트워크 가상화 에뮬레이터인 eve-ng에서 지원하는 오픈소스 트래픽 생성기로서 네트워크 상에 위치하는 drone(에이전트)이 제어 시스템의 설정값을 받아서 실제 패킷을 생성하는 시스템이다.

트래픽 발생 도구들은 트래픽 부하 측정에 목적을 두고 있으며 훈련을 위한 목적으로 개발되지 않아 실제 시스템에서의 사용자의 트래픽 양상을 재현하거나 L7 payload의 내용이 재현 상황에 적합한가는 고려하지 않는다. 본 논문에서 제안하는 상태 저장 트래픽 발생 Architecture는 훈련을 위한 적합한 트래픽을 생성하는 것에 초점을 맞추었다는 점에서 기존 트래픽 발생기들과 차별화된다.

3. 상태 저장 트래픽 제공 Architecture 설계

본 논문에서 제안하는 상태 저장 트래픽 발생(Stateful Traffic Generation) Architecture는 사이버전 훈련 시스템

에 배경 트래픽을 제공하기 위한 목적을 가진다. 이를 위해 제안 Architecture는 사이버전 훈련을 위한 네트워크 토폴로지를 구성하고 있는 가상머신에 다양한 종류의 배경 트래픽을 제공하는 에이전트와 각 가상머신에 설치된 에이전트를 통합 관리하기 위한 모니터링 서버로 구성된다. 이러한 운용 개념을 그림으로 나타내면 Fig. 1과 같다.

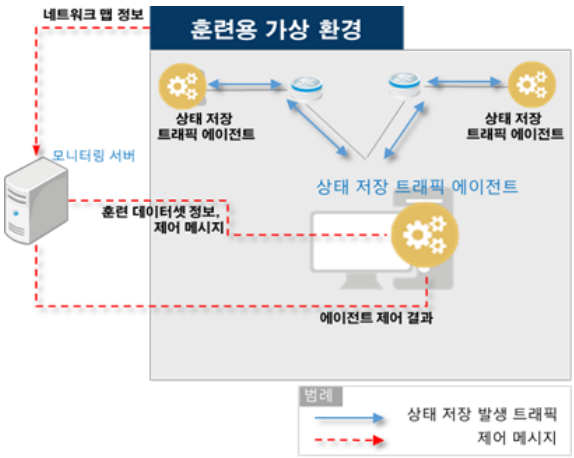

Fig. 1. Operational concept

훈련을 위한 트래픽 제공이라는 특성상 트래픽 제공 에이전트는 설치된 각 가상머신에 다양한 종류의 상태 저장 배경 트래픽을 제공하면서 해당 가상머신에 접속하여 사이버전 훈련을 수행하는 훈련자들의 네트워크 관련 행위를 방해하지 않아야 하는 조건을 가진다. 또한 다양한 네트워크 토폴로지를 가질 수 있는 훈련용 가상 환경을 지원하기 위해서 모니터링 서버는 트래픽 에이전트가 설치된 가상머신이 유동적인 상황을 고려하여야 한다. 각 필요 요소들의 세부 설계 내역은 다음과 같다.

3.1 상태 저장 트래픽 데이터셋 구성

사이버전 훈련을 위해 제공되는 배경 트래픽은 훈련의 대상이 되는 위협 트래픽을 숨기기 위한 목적을 가진다. 만약 배경 트래픽이 없을 경우, 훈련자는 트래픽 자체만을 감시하다 트래픽이 생길 경우 이를 차단하기만 하면 되기 때문이다. 이러한 목적의 배경 트래픽을 좀 더 실제와 유사하게 제공하기 위해서는 훈련의 목적이 되는 실환경에서 트래픽을 캡처하여 데이터셋(PCAP: Packet CAPtrue, 패킷 캡처) 형태로 제공하는 방법을 채택하였다. 본 절에서는 실환경에서 캡처된 트래픽을 훈련용 가상환경에서 상태 저장 방식으로 유통시키기 위해 수집된 실 트래픽 데이터셋을 가공하여 상태 저장 트래픽 데이터셋으로 변환하는 방법에 대해 설명한다.

가공 시 고려해야 할 요소는 다음과 같다.

- 상태 저장 방식으로 발생할 수 있도록 발생할 패킷 순서를 일반 packet의 TCP Header가 가지고 있는 sequence field와 다른 별도의 reserved field에 저장: 네트워크 통신 시, Router, switch 단에서의 변경 방지를 위함
- 훈련자의 네트워크 행위에 영향을 끼치지 않아야 함: 요구된 배경 트래픽 발생에 의한 패킷과 훈련자 행위에 의한 패킷 구분 필요
- 서로 다른 데이터셋(PCAP)이 서로 간에 영향을 주지 않아야 함: 배경 트래픽 간 패킷 구분 필요
- 실제와 유사한 트래픽 제공: L7 payload를 수정하지 말아야하는 제약 사항을 가짐

위의 사항을 고려할 때, L7 payload, 물리 interface인 Ethernet Header를 제외하고 패킷 구분 field와 sequence field를 지정할 수 있는 패킷 field는 IP header와 TCP/UDP header 부분이 해당된다. 그리고 상태 저장 데이터셋이라는 특성 상, 패킷의 순서를 준수해야하기 때문에 TCP protocol을 활용한다고 가정하고 데이터셋 변환을 수행한다.

패킷 구분 field는 훈련자 행위에 의한 패킷과 상태 저장 트래픽 발생에 의한 패킷의 구분을 위한 field와 배경 트래픽 간의 데이터셋 구분을 위한 field(이후 group id)를 별개로 간주하여 진행한다. 전자의 경우, IP header 중 패킷이 얼마나 빨리 처리되어야 하는가에 대한 정보를 담는 Type of Service(TOS) field에 특정값(0×40)을 지정하는 방식으로 수행한다. Group id는 하나의 PCAP file로 만들어지는 상태 저장 트래픽을 다른 PCAP 상태 저장 트래픽과 구분하기 위한 목적으로 TCP header의 window field(16 bits)를 활용하여 unique한 id를 부여한다. 또한 데이터셋 내부의 패킷들의 순서를 확인하기 위한 field는 URG 플래그가 설정되지 않으면 사용되지 않는 field인 Urgent pointer(16 bit) 부분을 활용한다. 이를 통해 만들어지는 packet의 header는 Fig. 2의 형태를 가지며 bit 단위로 정리한 표가 Table 1이다.

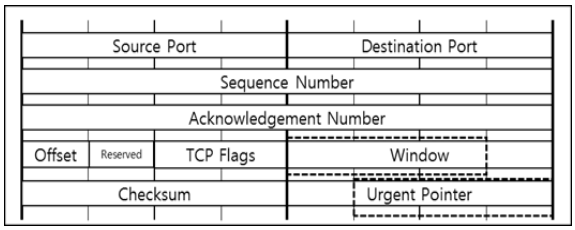

Fig. 2. Stateful traffic dataset header structure

Table 1. Stateful traffic dataset header

Layer	field	설정
IP	TOS	64 (0×40)
	Protocol	TCP
TCP	Window Size	[15:12] = 0
		[11:0] = 해당 패킷의 Group ID
	Urgent Point	[31] = 0 (수신 패킷)
		[30:16] = 해당 패킷의 Sequence Number

모니터링 서버는 훈련용 가상환경의 네트워크 맵 정보를 받아 트래픽 에이전트가 실행될 VM의 IP 정보를 추출하고 위에서 설명한 field 값을 함께 실환경 PCAP file에 적용하여 트래픽 에이전트가 실행할 PCAP 데이터셋을 생성한다.

3.2 상태 저장 트래픽 에이전트 설계

트래픽 에이전트는 3.1절에서 설명한 상태 저장 트래픽 데이터셋을 순서에 맞게 발생시키는 것을 목적으로 한다. 이를 위해 트래픽 에이전트는 설정된 데이터셋을 기반으로 Ethernet layer를 통과한 패킷을 interrupt하여 데이터셋에 의해 발생한 패킷인지 검사한 후, 아닐 경우, 원래 처리되어야 하는 Network layer에서 처리할 수 있도록 하는 내부 private firewall 개념을 적용하였다.

내부 private firewall에 의해 tos field가 0×40인 패킷을 식별하여 에이전트 처리 대상 패킷임을 인식한 후에는 Group ID에 의한 구분 및 sequence를 비교한 후, 데이터셋에 의해 다음에 보내야할 packet을 송신한다. 이 과정은 Group id field인 Window size 값에 의해 내부에서 관리하는 Packet Queue에 저장하는 과정과 Queue에 저장된 packet을 그 시점에서 받아야하는 순서의 패킷인지를 비교하여 다음 패킷을 발생시키는 과정으로 분리하여 비동기적으로 처리한다. 이 동작 매커니즘을 Fig. 3로 나타낼 수 있다.

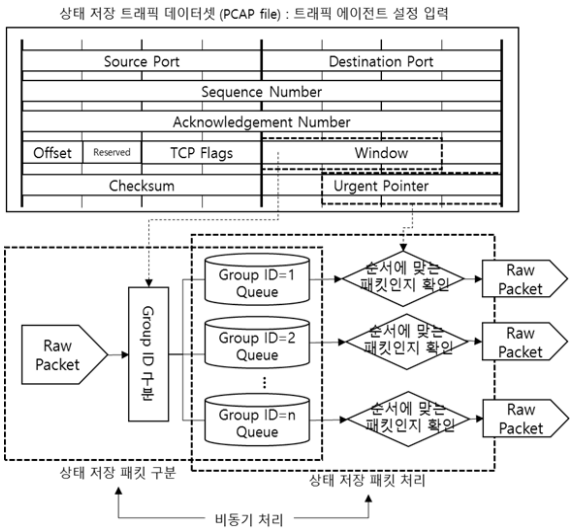

Fig. 3. Asynchronous stateful packet processing

트래픽 에이전트는 Ethernet layer 레벨에서 packet을 처리하기 때문에 TCP layer에 의한 재전송 메커니즘을 활용하지 않는다. 동시에 상태 저장 방식으로 트래픽을 발생시키기 때문에 만약 네트워크 환경에서 소실되는 패킷이 생길 경우, 다음 패킷으로 진행할 수 없게 된다. 이를 처리하기 위해 별도의 재전송 protocol을 정의하여 패킷 손실에 대한 대응 작업을 수행할 필요가 있다. 본 논문에서는 이를 위한 재전송 protocol로 다음과 같은 절차를 제안한다.

a. packet 송신 후, 상태 저장 트래픽 데이터셋에서 다음 패킷이 수신 패킷일 경우, 사전 정의된 timeout 시간만큼 패킷 수신을 대기한다.
b. timeout 시간 동안 수신이 실패하거나 받아야 하는 packet보다 sequence보다 큰 sequence를 가진 패킷을 받았을 경우, c로 이동한다. 재전송 요청 패킷을 받았을 경우, 요청된 패킷을 재발송한 후, a로 이동한다. 받아야하는 packet 수신이 성공했을 경우, a로 이동한다.
c. 수신 패킷을 발생시키는 트래픽 에이전트로 수신해야하는 패킷의 재전송 요청 패킷을 송신한다. 패킷 송신 후, 사전 정의된 timeout 시간만큼 패킷 수신

을 대기한다. 재전송이 무한히 되풀이되는 것을 막기 위해 5번의 재전송 패킷이 발송된 후에는 통신이 실패했음을 알리고 종료한다.

이에 대한 예시를 Fig. 4로 나타내었다.

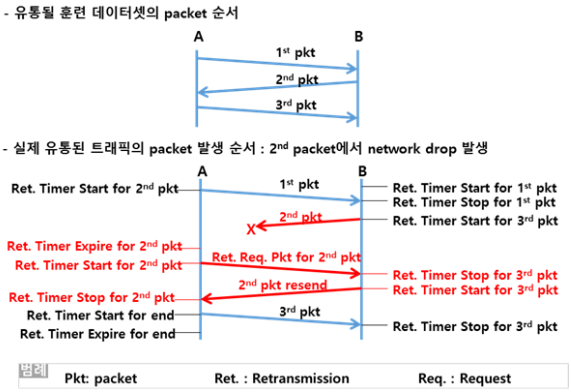

Fig. 4. Retransmission protocol

A는 packet 1을 송신하고 packet 2에 대한 재전송 timer를 설정된 timeout으로 작동시킨다. B는 트래픽 데이터셋 발생 시작 시각부터 packet 1에 대한 재전송 timer를 작동시킨 상태에서 받아야 하는 packet 1을 받고 packet 2를 A로 송신한다. 그러나 packet 2가 네트워크 상황에 의해 소실되어 A에 전달되지 못하는 경우, A는 packet 1 송신 후, 작동된 재전송 timeout이 지난 후 2번 packet을 못 받은 상태를 인식하고 2번 packet에 대한 재전송 요청 패킷을 B로 송신하게 된다. B는 2번 packet 재전송 요청 패킷을 받은 시점에서 3번 packet에 대한 재전송 timer를 중지하고 2번 packet을 재전송하게 된다. 그리고 다시 3번 packet에 대한 재전송 요청 timer를 작동시킨다.

재전송 요청 packet을 수신 측에서 식별하기 위해서는 재전송 요청 패킷임을 나타내는 flag를 별도의 field에 설정할 필요가 있다. 이를 위해 3.1절에서 제시한 Window size 혹은 Urgent Pointer field를 사용할 수 있다. 다른 field를 활용할 수도 있으나 하나의 packet을 처리하기 위해 접근해야하는 field의 종류를 늘리는 것은 불필요한 IO를 발생시킬 수 있으므로 위의 두 field를 재활용하는 방안으로 접근하였다. 또한 재전송 packet은 Group ID 식별 후, sequence를 비교하기 위한 프로세스를 처리해야할 필요가 없으므로 Group ID를 저장하는 window size field의 reserved bit를 활용하여 재전송 요청 packet임을 표시하고 Urgent Point에 재전송 요청 packet의 sequence number를 저장하는 방식으로 재전송 packet을 새로 생성하여 송신하도록 한다. 이를 위해 추가된 field를 포함한 상태 저장 트래픽 패킷의 TCP Header는 Table 2와 같다.

Table 2. Stateful traffic dataset header redefine

Layer	field	설정
TCP	Window Size	[15:12] = 0
		[12] = 1 재전송 요청 패킷
		[11:0] = 해당 패킷의 Group ID
	Urgent Point	[31] = 0 (수신 패킷)
		[30:16] = 해당 패킷의 Sequence Number

트래픽 에이전트는 설정된 상태 저장 트래픽 데이터셋 발생 시에 packet이 소실되는 경우가 발생할 경우, Windows size field 12번째 bit를 자체적으로 변경하여 재전송 요청 packet을 전송하게 된다.

3.3 모니터링 서버 구조 설계

모니터링 서버는 상태 저장 트래픽 데이터셋을 생성하고 이를 훈련용 가상 환경에 설치된 상태 저장 트래픽 에이전트에 배포하여 트래픽 발생을 할 수 있도록 제어하는 역할을 담당한다. 기능적인 부분으로 살펴보면 트래픽 데이터셋 생성과 트래픽 에이전트 제어 부분으로 크게 나눌 수 있다.

트래픽 데이터셋 생성은 훈련용 가상환경의 네트워크 토폴로지와 실환경 트래픽 캡처 데이터를 입력으로 받아 트래픽 에이전트에서 처리 가능한 형태의 데이터셋으로 변환하는 기능이다.

훈련용 가상환경은 외부 시스템에 의해 정의되는 환경으로서 상태 저장 트래픽 발생 Architecture는 네트워크 토폴로지를 구성하는 가상 머신의 정보와 네트워크 링크 정보를 외부 입력 데이터로서 받게 된다. 훈련용 가상환경에서 배경 트래픽을 송/수신할 가상 머신을 선택한 후, 가상 머신에서 발생할 트래픽을 실환경 캡처 트래픽 중에서 선택하면 3.1절에서 설명한 상태 저장 트래픽 데이터셋을 생성하게 된다. 각각의 외부 데이터를 어떻게 활용하는지를 Fig. 5로 나타내었다.

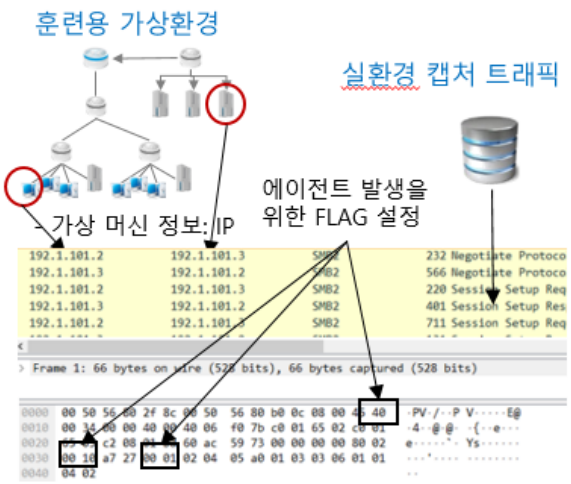

Fig. 5. Stateful traffic dataset generation

트래픽 에이전트 제어 기능은 훈련용 가상환경에 분산된 상태 저장 트래픽 에이전트들을 통합하여 관리하기 위한 기능이다. 분산된 에이전트를 관리하기 위해서는 기능적 요구사항과 함께 고려되어야 할 비기능적 요구사항, 즉 품질 속성[6]으로서 Table 3과 같은 항목이 존재한다.

Table 3. Quality attribute scenario

품질 속성	설명
확장성	상태 저장 트래픽 생성 Architecture는 다양한 네트워크 토폴로지를 가지는 훈련용 가상환경을 지원해야 한다.
정확성	각 가상 머신 local clock은 재전송 프로토콜에서 적용하는 타이머 주기 이상 차이를 가져서는 안 된다.
시간 반응성	트래픽 에이전트의 제어 명령 수행은 Global clock 기준으로 동일한 시점에 수행되어야 한다.

첫 번째 품질 요소인 확장성은 상태 저장 트래픽 발생 Architecture의 적용 대상이 훈련용 가상환경이라는 점에서 도출된다. 훈련용 가상환경은 외부 시스템에 의해 네트워크 토폴로지가 저작되고 가상환경(ex: VMware[1]) 상에 생성되며 훈련의 목적과 대상에 따라 다양한 형태의 네트워크 환경을 가지게 된다. 이러한 다양한 네트워크 토폴로지에서는 트래픽 에이전트가 설치될 가상 머신의 설치 대수가 가변적일 수 밖에 없으며 각 머신이 가지는 IP 주소 역시 가변적인 요소가 된다. 이에 대해 모니터링 서버가 각각의 모든 트래픽 에이전트에 session을 유지하며 제어 명령을 전달하는 것은 불필요한 성능적 부하로 작용할 수 있으므로 제어 메시지는 publish/subscribe 구조[6]로 모니터링 서버와 트래픽 에이전트 간의 직접적인 연결 없이 전달될 수 있는 형태로 설계하였다.

트래픽 에이전트는 가상머신에서 동작 시, 제어 메시지 통신을 위한 별도의 네트워크 어댑터 모듈을 추가하여 해당 어댑터에서만 제어 메시지를 처리하도록 한다. 이를 그림으로 나타내면 Fig. 6과 같다.

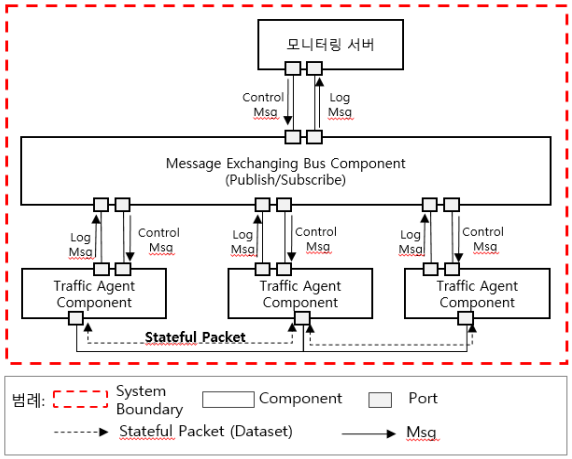

Fig. 6. Monitoring server – traffic agent

두 번째 품질 속성인 정확성은 시간 동기화에 대한 제약을 부여하는 항목으로 3.2절에 설명한 트래픽 에이전트의 재전송 protocol에 의해 주어진 품질 속성이다. 트래픽 데이터셋에 의해 송/수신하기로 설정된 트래픽 에이전트 간의 local time이 상호 간에 타이머 이상 차이가 난다면 주어진 트래픽 발생 시각을 local time이 빠른 쪽이 먼저 시작하게 되어 송신 혹은 수신되어야 할 packet이 timeout 시간까지 처리되지 않아 필수적으로 재전송 요청 packet이 발생하기 때문이다. 네트워크 환경 상의 문제가 아닌 상황에서 재전송 요청 packet을 발생시키지 않게 하기 위해 각 가상머신의 local clock을 기준 global clock을 정해 동기화시킬 필요가 있다. 모니터링 서버의 시각을 global clock으로 설정하고 트래픽 에이전트의 초기화 시에 반드시 동기화를 수행하도록 하는 방식으로 2번 품질 속성을

만족시킬 수 있다. 이러한 시간 동기화를 위한 방법으로는 네트워크 타임 프로토콜(Network Time Protocol: NTP)을 일반적으로 사용하며 인터넷에 연결되어 있을 경우에는 time.google.com 등 공용 NTP 서버와의 동기화를 수행할 수 있다. 그러나 훈련 시스템은 대부분의 경우 외부와의 연결이 없는 폐쇄망이므로 모니터링 서버를 내부 NTP 서버로 설정하고 각각의 훈련용 가상환경 내의 가상 머신들이 모니터링 서버를 기준으로 동기화하도록 시스템 설정을 수행해야 한다.

마지막 품질 속성인 시간반응성은 첫 번째 품질 속성에 의한 설계와 두 번째 품질 속성에 의한 설계에 따라 파생된 속성이다. 확장성을 만족하기 위해 채용한 publish/subscribe 구조에서는 메시지 소비자가 메시지를 polling해가는 시점이 달라 명령 처리에 대한 시간 동기화를 수행할 수 없다는 점과 트래픽 에이전트가 같은 트래픽 데이터셋을 송/수신하기 위해서는 재전송 타이머 시간 범위 내에서 시간 동기화를 이루어야 한다는 점이 충돌하는 것이다. 이를 해결하기 위해서 트래픽 에이전트가 메시지를 소비하는 시점과 실제로 제어 명령을 수행하는 시점을 분리하는 방식으로 설계하여 적용하였다. 이는 제어 명령 내부 정보에 제어 명령을 수행할 시각을 함께 주어 모든 트래픽 에이전트가 global clock 동기화를 수행한 local clock의 특정 시점에 제어 명령을 수행하도록 하는 것이다.

4. 상태 저장 트래픽 Architecture 구현 및 시험

3장에서 기술한 상태 저장 트래픽 발생기 설계 사항들을 기반으로 개발한 각 컴포넌트들을 정리하면 Table 4와 같다.

Table 4. Stateful traffic generation components

컴포넌트	역할
메시지 교환 버스	Master/Slave agent간 통신을 위한 publish/subscribe 구조 지원
모니터링 서버	상태 저장 트래픽 데이터셋 생성 트래픽 에이전트 제어 내부 NTP Server
트래픽 에이전트	훈련용 가상환경에서 상태 저장 트래픽 발생

각각의 컴포넌트들의 시스템 상 배치를 Fig. 1에 추가하여 구성하면 Fig. 7과 같다.

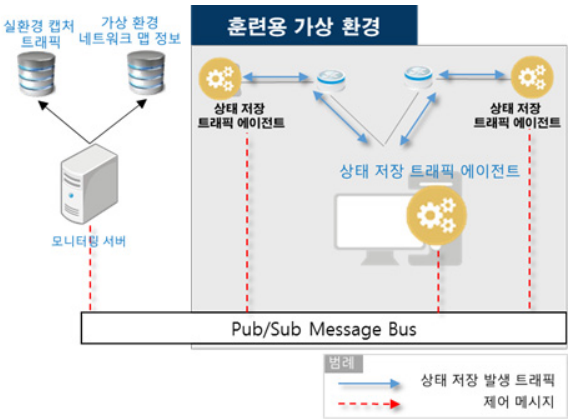

Fig. 7. System architecture

4.1절에서는 3장의 설계 내역을 바탕으로 정리된 각각의 컴포넌트들에서 구현 형태를 설명하고 4.2절에서는 예제 시나리오 구동 시, 캡처된 실유통 트래픽과 생성한 상태 저장 트래픽 데이터셋을 비교한 내용을 구현 결과로 제시한다.

4.1 Architecture Component 구현

4.1.1 메시지 교환 버스

확장성을 위해 선택한 publish/subscirbe 메시지 교환을 위한 메시지 교환 버스(message exchanging bus)는 직접 구현하는 대신 분산 시스템 메시지 교환 중 대규모 시스템에 적합한 Kafka 서버를 사용하여 연결한다. Apache Kafka는 publish/subscribe 구조을 채용한 분산 메시징 시스템으로 LinkedIn에서 개발하여 2011년에 오픈소스로 공개된 상태이다[4]. Kafka는 관리할 메시지를 topic을 기준으로 관리하며 소비자가 특정 메시지를 받고자 할 경우, 해당 메시지가 가지고 있는 topic name을 통해 메시지를 소비하겠다고 선언하도록 형식으로 동작한다.

4.1.2 모니터링 서버

모니터링 서버는 Ubuntu 16.04 LTS OS를 운영체제로 가지며 내부 NTP server로서의 기능은 리눅스의 ntp daemon의 기능을 활용하여 제공한다. 모니터링 서버는 훈련 시작 전에 Fig. 7에 나타난 실환경 캡처 트래픽 DB와 가상환경 네트워크 맵 정보 DB의 저장 정

보를 사용하여 상태 저장 트래픽 데이터셋을 PCAP file 형태로 생성한 후, Kafka server의 Topic을 활용하여 이를 트래픽 에이전트에 배포한다. 모니터링 서버는 데이터셋 배포를 완료한 후, 메시지 교환 버스를 통해 트래픽 발생 시작, 중지와 같은 제어 명령을 트래픽 에이전트를 전파하여 유통 상태를 관리한다.

4.1.3 트래픽 에이전트

트래픽 에이전트는 운영체제 Linux와 Windows를 지원하며 모니터링 서버가 Kafka server를 통해 전달한 상태 저장 트래픽 데이터셋 PCAP file을 제어 명령에 의해 발생 시작, 중지하는 기능을 수행하기 위해 Fig. 8과 같은 구조로 구현하였다.

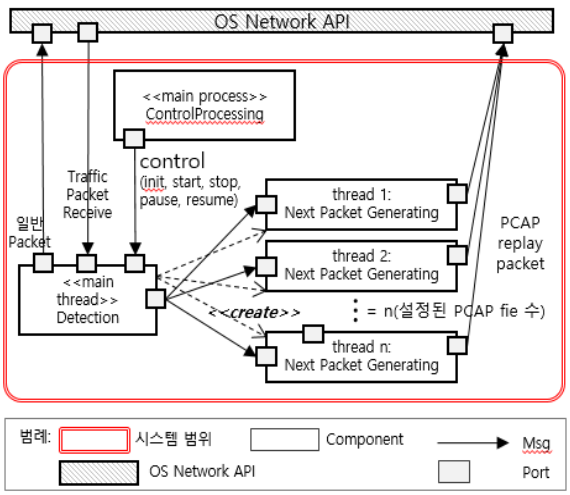

Fig. 8. Processes in traffic agent

시스템 범위에서 main thread로 표시된 Detection component는 3.2절에서 설명한 private firewall 역할을 수행하며 하나의 트래픽 에이전트 당 하나의 thread로 활성화되는 구조이다. 가상 머신 외부에서 들어온 트래픽을 네트워크 카드가 처리하여 Ethernet layer가 제거된 packet이 TCP/IP layer로 전달되기 전에 데이터셋에 의해 송/수신되는 패킷인지 확인하기 위해 OS 커널단에 API를 통해 접근할 필요가 있다. 트래픽 에이전트 구현 시에는 Linux 버전은 Netfilter(Linux 커널의 네트워크 스택에 있는 패킷 필터링 Hook)를, Windows 버전은 Windivert(Windows Packet Divert: 윈도우 네트워크 스택에서 전송된 네트워크 패킷을 필터링 규칙에 따라 캡처/수정/폐기할 수 있는 기능을 지원하는 라이브러리)를 사용하여 커널 단에 접근하였다. Detection module은 데이터셋 발생 패킷일 경우, group_id를 보고 그 group_id를 가진 PCAP file packet을 처리하는 thread로 해당 packet을 전달한다. 전달하는 방식은 두 개의 thread가 공유하고 있는 queue에 packet을 push/pop하는 방식으로 이루어지며 push/pop 하는 시점에서 signal을 발생시켜 수신과 처리가 비동기 방식으로 운용될 수 있도록 구성하였다. 세부 로직은 Fig. 9의 pseudo code로 나타내었다.

```
Detection module:
    def compare(pkt):
        if pkt.ip_header.tos == ox40:
          retrams_flag = pkt.tcp_header.window_size[12]
          if retrans_flag == True:
              send_pkt(PCAP_file[req_seq])
              return False
          group_id = pkt.tcp_header.window_size[11:0]
          Queue_item[group_id].put(pkt)
          send_signal(thread_item[gropu_id])
          return False
        else:
           return True

Next_Packet_Generating module:
    def processing():
        while PCAP file is not END:
            ptk = PCAP.read(position)
            if pkt.src == self
                send_pkt(pkt)
            else:
                flag = wait_signal(time_out)
                if flag == False:
                  retrans_pkt = make_retrans_req(pkt)
                  send_pkt(retrans_pkt)
```

Fig. 9. Traffic agent pseudo code

4.2 상태 저장 트래픽 발생기 시험

상태 저장 트래픽 발생기의 모니터링 서버와 트래픽 에이전트가 설계 의도와 동일하게 동작하는지를 검증하기 위하여 Fig. 10의 시스템 구성도로 가상 머신 네트워크를 구성하여 실험을 수행하였다. 이 가상

운용환경은 총 28개의 가상 머신으로 이루어져 있으며 각각의 가상 머신은 가상 스위치와 라우터를 이용하여 연결되어 있는 구조이다.

트래픽 에이전트는 총 28개의 가상 머신(OS: Ubuntu 16.04, Windows 10)에서 동시에 트래픽을 발생하여 트래픽 데이터셋의 송/수신 노드에 따라 패킷 송/수신을 수행하였다. 이 중 하나의 가상 머신에서 wireshark를 사용해 가상 머신의 송/수신 packet을 capture한 결과는 Fig. 11과 같다.

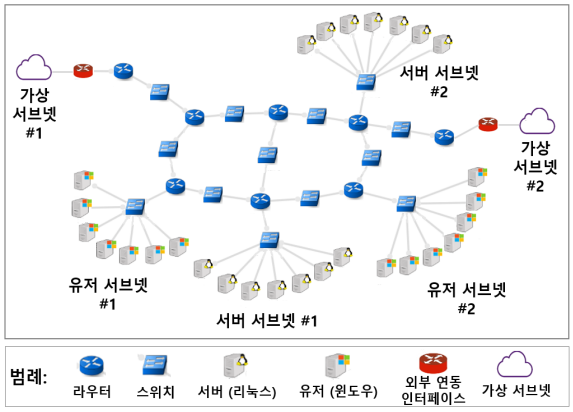

Fig. 10. Virtual environment network topology

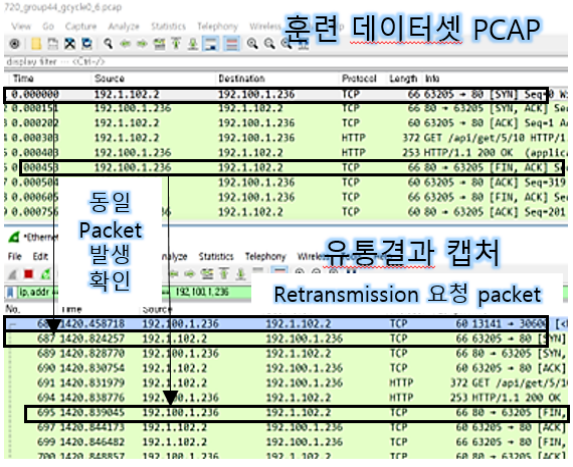

Fig. 11. Comparison between dataset traffic and captured traffic

Fig. 11의 설명과 같이 상단 데이터셋 트래픽의 양상과 하단 실유통 캡처 트래픽이 패킷 별로 대조하였을 때 동일한 순서와 형태로 나타난 것을 확인하여 데이터셋 트래픽 유통 및 트래픽 에이전트 제어가 설계한 대로 동작함을 확인할 수 있었다.

5. 결 론

본 논문에서는 실시간으로 진화하는 사이버 위협에 대응하는 시스템 보안 담당자의 역량을 향상시키기 위한 사이버전 훈련 시스템에 배경 트래픽을 유통시킬 수 있는 상태 저장 트래픽 발생 Architecture의 설계와 구현 결과를 제시하였다. 이 Architecture는 기존의 하드웨어, 소프트웨어 트래픽 발생기를 활용한 배경 트래픽 발생과는 다르게 실제 유통된 트래픽을 어떤 훈련용 가상 환경에서든 재활용할 수 있도록 상태 저장 트래픽 데이터셋으로 변환하는 기능을 지원하며, 트래픽 에이전트에서 각각의 packet들이 수신됨을 확인하고 송신하는 방식을 취함으로써 훈련자가 트래픽을 분석할 때 의미를 찾을 수 없는 packet의 나열이 아니라 session 내에서 이루어지는 트래픽 플로우로서 분석할 수 있도록 한다.

상태 저장 트래픽 발생 Architecture는 훈련용 가상 환경에 훈련자에 의해 발생하는 트래픽 뿐 아니라 실 환경에서 발생하는 트래픽을 유통시킬 수 있는 방안으로서 사이버전 훈련자로 하여금 좀 더 현장감 있는 사이버전 훈련을 수행할 수 있도록 지원한다. 이 Architecture의 결과물을 활용하여 훈련자는 방어 대상 시스템의 송/수신 트래픽을 분석하여 정상적으로 유통되는 트래픽과 섞여있는 위협 트래픽을 식별할 수 있는 훈련을 수행할 수 있다. 향후, 훈련 대상 실제 시스템에서 유통되는 트래픽을 수집한 데이터셋을 이용하여 훈련용 가상 환경에 유통될 트래픽 계획을 저작할 수 있도록 유통계획 저작 인터페이스를 개발하고 이를 사이버 훈련 시스템에서 사용할 수 있도록 발전시켜야 한다. 또한 이 Architecture를 효율적으로 활용하기 위해서는 훈련 대상 실제 시스템에서 유통되는 트래픽을 수집하여 실제 상황의 트래픽이 어떠한 양상을 보이는지에 대한 분석하는 연구가 병행되어야 할 것이다.

후 기

이 논문은 민·군기술협력사업의 지원으로 수행된 연구임(UM17312RD3).

References

[1] Myung Kil Ahn, Yong Hyun Kim, “Research on System Architecture and Simulation Environment for Cyber Warrior Training,” Journal of the Korea Institute of Information Security & Cryptology, Vol. 26, No. 2, pp. 533-540, 2016

[2] B. Ferguson, A. Tall, and D. Olsen, “National Cyber Range Overview,” Proceedings of the 2014 IEEE Military Communications Conference, MILCOM ’14, pp. 123-128, Oct. 2014

[3] T. Bonaci and J. Herron and T. YusufTo, “Make a Robot Secure: An Experimental Analysis of Cyber Security Threats Against Teleoperated Surgical Robotics,” National Science Foundation, CNS-132975 1, pp. 1-11, May 2015.

[4] Suyoun Hong, Kwangsoo Kim, Taekyu Kim, “The Design and Implementation of Simulated Threat Generator based on MITRE ATT&CK for Cyber Warfare Training”, Journal of the KIMST, Vol. 22, No. 6, pp. 797-805, 2019.

[5] Soumya Mahalakshmi A., Amulya B. S. and M. Moharir, “A Study of Tools to Develop a Traffic Generator for L4 - L7 Layers,” 2016 International Conference on Wireless Communications, Signal Processing and Networking(WiSPNET), Chennai, pp. 114-118, 2016.

[6] Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice,” Addison-Wesley Professional, America, 2012.