네트워크 보안 평가를 위한 유연한 테스트베드 설계

(A Design of Flexible Testbed for Network Security Evaluation)

임 이 진 † 최 형 기 †† 김 기 윤 †††
(Yi Jin Im) (Hyoung-Kee Choi) (Ki Yoon Kim)

요 약 본 논문에서는 보안장비의 성능평가 및 네트워크 내 센서들의 로그정보를 수집할 수 있는 테스트베드를 구축하였다. 이 테스트베드는 실제 인터넷과 유사한 테스트 환경을 제공하여 테스트베드 내에서 공격을 직접 생성하거나 공격 트래픽이 포함된 데이타셀을 이용하여 공격을 재현할 수 있도록 구성되었다. 본 테스트베드는 기존 테스트베드에 비해 비교적 적은 비용과 시간으로 구축이 가능하며, 공격 트래픽의 유형이나 테스트베드 사용목적에 따라 수정이나 확장이 용이하다. 따라서 많은 비용과 시간의 소모로 인해 쉽게 진행할 수 없었던 보안장비의 성능평가나, 공격 발생 시 네트워크에 존재하는 센서들의 로그 수집을 용이하게 할 수 있다. 본고에서는 테스트베드 구축 시 발생할 수 있는 다양한 문제점과 그 해결방안을 제시하였으며 제안한 테스트베드를 이용하여 DDoS 공격과 웜을 재현하는 과정을 보였다.

키워드 : 테스트베드, 네트워크 보안, 데이타셀, 공격

Abstract We present a testbed for collecting log information and evaluating network security under various attacks. This testbed is modeled on real Internet, where attack traffic coexists with normal traffic. Attacks can be produced either by attack tools directly or by data sets including attack traffic. It costs less time and money than existing ones which are both costly and often time consuming in constructing. Also, it can be easily revised or extended according to the traffic types or the uses. Therefore, using our testbed can make various tests more efficient and facilitate collecting log information of sensors with attacks. We discuss how to use our testbed through replay procedures of DDoS attack and worm. We also discuss how we surmount some difficulty in constructing the testbed.

Key words : testbed, network security, dataset, attack

1. 서 론

네트워크에는 위협적인 공격이 증가하고 있다. 2006년 3월에 발표된 시만텍 보고서[1]에 따르면 조사 기간 동안 하루 평균 서비스 거부 공격의 발생 횟수는 전년도보다 약 51% 증가한 수치를 보이고 있다. 네트워크 공격의 증가에 따른 피해를 줄이기 위해 국가 및 연구소, 기업 차원에서 다양한 정책 제시 및 연구를 수행하고 제품을 출시하여 증가하는 네트워크 공격에 대응하고 있다. 상용 제품들은 많은 공격에 대응하고 시스템을 보호할 수 있지만, 침입탐지시스템의 공격 오탐지로 인한 성능 저하나 알려지지 않은 공격에 대해서는 유연하게 대처하지 못하는 문제 등을 가지고 있다. 이러한 문제점들의 대부분은 제품에 대한 성능평가나 충분한 검증을 거치지 못하는 것에서 기인한다.

보안 제품의 성능평가 및 검증을 위해서는 실제 네트워크에서 발생하는 공격에 대한 테스트가 요구된다. 그러나 실제 네트워크에서 테스트를 진행하는 것은 매우 위험하다. 만약 실제 네트워크에서 보안 제품의 성능평

† 정 회 원 : 성균관대학교 정보통신공학부
yjim@hit.skku.edu
†† 정 회 원 : 성균관대학교 정보통신공학부 교수
hkchoi@ece.skku.ac.kr
(Corresponding author임)
††† 정 회 원 : (주)파이오링크
doogy@piolink.com
논문접수 : 2008년 8월 28일
심사완료 : 2009년 10월 26일

정보과학회논문지: 정보통신 제37권 제1호(2010.2)

가를 위해 공격 트래픽을 유발할 경우 공격이 다른 네트워크로 전파되거나 많은 인터넷 사용자들에게 피해를 입힐 수 있다. 이를 방지하기 위해서는 실제 네트워크와 유사한 환경에서 테스트를 진행할 수 있는 독립된 네트워크가 필요하다. 실제 네트워크와 동일한 환경에서 테스트를 거치면 사용 중 발생 가능한 문제점에 대한 발견 및 수정이 가능하며 제품 성능의 검증이 가능하다.

테스트베드는 실제 인터넷과 유사한 환경에서 공격 트래픽 생성과 같은 위험한 실험을 수행할 수 있는 독립된 네트워크를 제공한다. 테스트베드를 이용한 실험에서는 네트워크에 나타나는 공격과 유사한 공격을 발생시켜 실제 인터넷상에서 공격이 발생했을 때와 유사한 정보를 얻을 수 있고, 로그 수집이나 보안 장비의 성능평가도 가능하다. 가상의 네트워크에서 공격 트래픽을 생성하여 실험하는 어플리케이션들도 있지만 실제로 존재하는 네트워크상의 센서 로그 수집이나 네트워크 보안 장비의 성능평가를 수행하기에는 어려움이 따른다.

테스트베드를 이용하여 공격을 생성시키는 방법으로는 1) 공격 도구를 이용하여 테스트베드 내에서 직접 공격을 발생시키는 방법과 2) 네트워크에 나타나는 트래픽을 시간의 흐름에 따라 배열한 데이타셀으로부터 공격을 재생하는 두 가지 방법이 있다. 공격 도구를 이용하여 공격을 발생시킬 경우에는 테스트베드 내에 공격 트래픽 이외의 정상적인 트래픽이 나타나지 않으며, 공격 트래픽과 정상적인 트래픽이 공존하는 인터넷에서의 실험과 유사한 결과를 얻기 위해서는 정상적인 트래픽도 생성시켜야 한다는 어려움이 있다. 반면, 데이타셀을 이용하여 트래픽을 재현할 경우 데이타셀에 포함된 공격 트래픽과 정상적인 트래픽 모두 발생시킬 수 있지만 데이타셀으로 제공되는 것 이외의 공격 트래픽을 생성할 수 없다는 단점이 있다.

본 논문에서는 실제 인터넷을 모델로 하며 실험의 목적에 따라 환경 설정을 변경할 수 있는 테스트베드를 제안한다. 제안하는 테스트베드는 기존의 테스트베드에 비해 비교적 적은 비용과 시간으로 구축이 가능하다. 공격 재현 시, 공격도구를 이용하여 공격을 직접 발생시키는 방법과 데이타셀을 이용하여 공격이 포함된 트래픽을 재현하는 방법 모두를 사용 가능하도록 구성하였고 사용 목적에 따라 확장이나 수정가능 하도록 구성하였다. 또 네트워크에 존재하는 다양한 센서들을 직접 배치하여 공격 발생시 센서로부터 얻을 수 있는 패킷 정보나 로그를 얻을 수 있도록 구성하였다. 따라서 다양한 보안제품의 성능평가나 관련 연구에 사용이 용이하며 공격도구를 이용한 공격과정의 분석이나 네트워크에 존재하는 센서들의 로그를 이용한 공격 분석 및 탐지에 관한 연구에도 활용이 가능하다.

본 논문은 다음과 같이 구성되었다. 2장은 관련연구로써 다른 연구에서 사용된 테스트베드에 대한 소개와 장단점을 설명하며 3장에서는 제안하는 테스트베드의 구성과 특징에 대해 소개한다. 4장에서는 테스트베드를 이용한 slammer 웜과 DDoS 공격의 재현 과정을 소개하며 5장에서는 테스트베드를 구축하면서 발생한 문제점과 해결방법을 소개한다. 마지막으로 6장에서는 결론을 맺는다.

2. 관련연구

기존에 진행된 보안제품의 성능평가나 네트워크 공격에 관한 연구 중 테스트베드를 이용한 대표적인 연구로는 MIT Lincoln Lab에서 수행한 Evaluating Intrusion Detection Systems 연구[2]와 Benzel과 그의 동료가 수행한 Cyber Defense Technology Experimental Research project[3]가 있으며, 이러한 테스트를 수행할 수 있는 테스트베드의 구성에 관한 연구로 Gautam와 그의 동료가 수행한 A Testbed for Quantitative Assessment of Intrusion Detection Systems using Fuzzy Logic 연구[4]가 있다.

2.1 MIT Lincoln Lab

MIT Lincoln Lab은 1998년에서 2000년까지 수행한 Intrusion Detection Evaluation 연구[2]에서 공격을 포함한 네트워크 트래픽을 생성할 수 있는 테스트베드를 구성하였다. 그림 1은 MIT Lincoln Lab의 테스트베드 구성을 나타낸다.

구성한 테스트베드는 공격의 대상이 되는 내부 네트워크와 인터넷을 모델로 공격이 발생하는 외부 네트워크로 구성되었으며, 두 네트워크는 라우터를 이용하여 연결하였다. 내부 네트워크에는 공격의 대상이 되는 공격대상, 내부 네트워크에서 발생하는 트래픽을 생성해주는 내부 트래픽 생성자, 내부 네트워크의 트래픽을 모니터링 하는 내부 관찰자, 공격자 역할을 수행하는 내부 공격자가 있다. 공격대상은 4대의 컴퓨터로 이루어져 있으며 Linux, SunOS, Solaris, Windows NT 운영체제가 공격대상에 각각 설치되었다.

내부 트래픽 생성기는 가상머신을 이용하였다. 가상머신은 한 대의 컴퓨터가 마치 여러대의 컴퓨터처럼 동작하는 것과 같이 보이게 하는 기술로, 주로 운영체제의 커널이나 어플리케이션을 사용해서 구현한다. MIT Lincoln Lab에서는 한대의 가상머신을 이용하여 20대의 PC와 워크스테이션이 동작하는 것과 같은 효과를 보였다. 가상머신에서 사용한 트래픽은 수정된 커널 기반의 Linux상에 수정된 버전의 sendmail, telnet, login, 트래픽 생성 툴을 이용하여 생성하였다. 내부 관찰자는 tcpdump를 이용하여 트래픽을 수집하였고 16 GB의 SCSI 디스크를 저장장치로 이용하여 대용량의 네트워크 트래

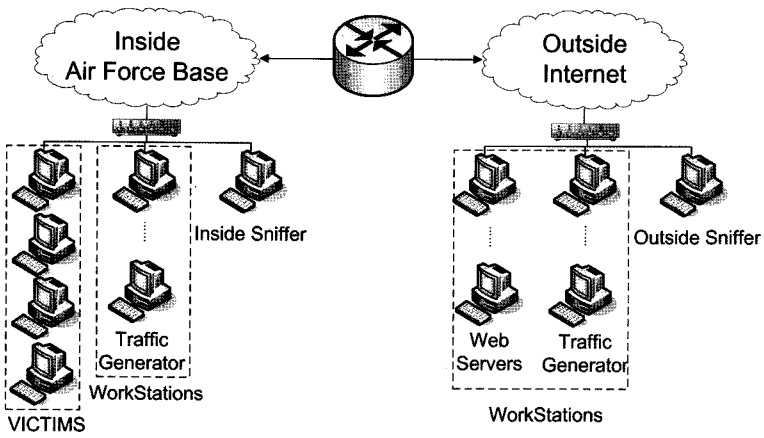

그림 1 MIT Lincoln Lab 테스트베드 구성

픽을 파일을 저장하였다. 내부 공격자는 두 대의 워크스테이션을 이용하였는데, 한 대는 Linux 운영체제를 설치하고 트래픽 생성 툴을 이용하여 자동적으로 공격이 발생되도록 하였으며, 다른 한 대는 Windows NT 운영체제와 PERL을 설치하여 자동적으로 공격을 수행할 수 있는 프로그램을 실행하였다.

외부에는 수천 대의 웹 서버 역할을 하는 인터넷 웹 서버, 외부에서 발생하는 트래픽을 생성해주는 외부 트래픽 생성자, 외부의 트래픽을 모니터링 하는 외부 관찰자, 공격자 역할을 수행하는 외부 공격자가 있다. 인터넷 웹 서버는 웹 서버 프로그램들을 표현해주는 것으로, 2500여대 호스트로부터의 요청에 대한 응답을 생성해주기 위해 수정된 웹 서버 프로그램을 이용하여 요청을 처리하도록 구성하였다.

MIT Lincoln Lab의 테스트베드는 트래픽 생성자를 이용하여 직접 트래픽을 생성하였고, 여러 운영체제와 프로그램을 사용하는 공격대상을 활용하였으며, 실제 인터넷과 유사한 환경을 제공하기 위해 많은 정상적인 트래픽도 생성하였다는 장점이 있다. 하지만 많은 수의 장비를 사용했으며, 가상머신을 구성하기 위해 Linux의 커널을 수정하고 사용하는 어플리케이션들도 수정해야 하므로 유사한 테스트베드를 구성하기에는 많은 시간과 비용이 요구된다는 단점이 있다.

2.2 DETER 테스트베드

Department of Homeland Security와 National Science Foundation 지원사업의 연구로 진행된 Cyber Defense Technology Experimental Research에서 제안된 테스트베드[3]이다. Jelena과 그의 동료들이 진행한 연구[5]에서는 시그니처 기반의 침입탐지시스템인 Snort를 비롯한 7개의 침입탐지시스템의 성능평가를 위해 사용되었다. 이 밖에도 약 50개의 연구에 DETER 테스트베드가 사용되고 있다[20].

테스트베드는 VLAN으로 구성되어 있으며 USC ISI와 UC Berkeley, 두 클러스터를 포함한다. 테스트베드는 각 클러스터별 혹은 실험자가 정의한 노드셋 별로 테스트를 수행할 수 있도록 구성되어 있다. USC ISI는 64대의 머신, UC Berkeley는 32대의 머신으로 구성되어 있으며 인터넷을 통해 접근하여 사용할 수 있다. 인터넷을 통해 테스트베드에 접근하기 위해서는 인터넷과 테스트베드를 연결하는 user server에 사용자 등록을 해야 한다. 사용자 등록은 deter 테스트베드 관리자 사이트에 테스트베드 사용목적과 실험 내용등을 기재하여 신청한다.

등록된 사용자는 접근가능한 계정과 사용할 테스트베드 내의 머신을 할당받는다. User server는 사용자 인증과 머신할당에 대한 로그와 테스트베드 동작에 대한 로그등을 기록하는 역할도 함께 수행한다. 계정과 머신을 할당받은 사용자는 실험에 필요한 소프트웨어를 직접 설치한다. 그리고 물리적으로 테스트베드에 접근할 수 없기 때문에 할당받은 머신들을 이용하여 논리적인 네트워크를 구성하여 원격지에서 테스트를 진행하고 결과를 수집한다. 실험 종료후에는 사용된 머신들을 초기화한다.

Jelena과 그의 동료들이 진행한 연구[5]에서는 총 40대의 머신을 이용하였으며 공격 트래픽과 정상적인 트래픽 모두 직접 생성하는 방법을 사용하였다. 할당받은 머신들을 크게 두개의 노드셋으로 구분했으며 각 노드셋은 정상적인 사용자와 공격자를 모두 포함하고 있다고 가정했다. 두개의 노드셋을 내부 네트워크와 외부 네트워크로 구분하였으며 내부 네트워크에 침입탐지시스템을 설치하여 성능평가 실험을 진행하였다.

이 테스트베드의 장점으로는 1) 대규모 네트워크를

구축하였으며 데이타셋을 이용하여 공격을 재생하거나 공격 도구를 이용하여 직접 공격을 실행할 수 있다는 것과 2) 등록된 사용자들은 직접 테스트베드를 구축하지 않아도 인터넷을 통해 원격지에서 테스트베드의 사용이 가능하다는 점이 있다. 그러나 많은 장비를 이용하여 테스트베드를 구축하였기 때문에 비용이 많이 소모되며, 소프트웨어가 아닌 하드웨어 장비를 직접 설치하여 수행하는 실험의 경우 사용하기 적절치 못하다는 단점이 있다. 또, 많은 사용자가 동시에 테스트베드를 사용할 경우 다른 사용자의 실험으로 인해 실험 결과에 영향을 받을 수 있고 장치의 충분한 사용이 제한될 수도 있다는 단점이 있다.

2.3 TIDeS

Gautam과 그의 동료는 논문[4]에서 침입탐지시스템의 성능평가를 위한 테스트베드를 제안하였다. 테스트베드는 총 4대의 머신으로 구성된다. 공격의 대상이 되는 2대의 공격대상머신과 사용자에게 테스트베드의 트래픽 재생이나 실험진행 상태를 확인하고 제어할 수 있는 인터페이스를 제공해주는 컨트롤머신 1대, 여러 호스트가 존재하는 것과 같은 효과를 제공해 주는 1대의 가상호스트 머신이 존재한다.

컨트롤 머신은 가상호스트 머신을 제어하여 트래픽을 발생시키거나 중지할 수 있다. 가상호스트 머신은 테스트베드내에 여러 호스트가 존재하는 것과 같은 효과를 나타내기 위해 가상머신 프로그램에 의해 여러 IP가 할당되어 동작하며, 정상적인 트래픽과 공격 트래픽을 생성한다. 이 때 인터넷으로부터 공격이 포함된 데이타셋을 수집하여 트래픽을 재현하는 것이 아니라 트래픽 생성 도구와 네트워크 공격 도구를 가상머신에 설치하여 직접 트래픽을 생성한다. 생성되는 공격 트래픽은 가상호스트머신에서 공격대상머신으로 전송되며 그 사이에 침입탐지시스템과 같은 보안장비를 설치하여 성능평가를 수행할 수 있도록 구성되었다.

이 테스트베드는 총 4대의 머신으로 구성되었기 때문에 적은 수의 장비로 유사한 테스트베드 구축이 가능하며, 테스트베드내에 침입탐지시스템과 같은 장비를 원하는 위치에 추가 배치하여 실험할 수 있다는 장점이 있다. 또, 실제 인터넷과 유사한 환경을 제공해주며 트래픽을 직접 생성하기 때문에 원하는 공격을 생성하여 테스트 할 수 있다는 장점도 있다. 하지만 트래픽을 직접 생성하기 때문에 새로운 공격 트래픽을 이용한 실험을 수행하기 위해서는 새로운 공격을 발생시키기 위한 프로그램의 구현이나 도구의 수집이 요구된다. 또 모든 트래픽을 가상호스트 머신에서 생성하는 구조로 되어 있기 때문에 네트워크에 존재하는 라우터와 같은 역할을 수행하는 노드가 없다. 따라서 생성되는 패킷들이 라우터를 거쳐 전송되는 형태가 아니라 같은 네트워크상에서 전달되기 때문에 라우터에 설치하여 사용하는 방화벽과 같은 장비는 사용이 불가능하다. 그러므로 라우터나 방화벽과 같은 장비의 로그 수집이나 성능평가를 수행할 경우에는 적합하지 않다.

3. 제안하는 테스트베드 소개

이 장에서는 본 논문에서 제안하는 테스트베드를 소개한다. 이 테스트베드는 적은 수의 장비로 실제 인터넷과 유사한 환경을 제공하며, 트래픽 생성 시 공격 트래픽을 직접 생성하거나 데이타셋을 이용해 재현할 수 있도록 하였다. 테스트베드를 구성하는 하드웨어는 5대의 PC로 구성되며, 소프트웨어는 패킷 재생 어플리케이션과 침입탐지시스템, 방화벽, 네트워크 모니터링 어플리케이션으로 구성된다.

3.1 하드웨어 구성요소

그림 2는 테스트베드의 하드웨어 구성을 보여준다. 테스트베드는 공격 트래픽이 포함된 트래픽이 생성되는 외부와 생성된 트래픽의 목적지 및 공격의 대상이 존재하는 내부로 구성된다. 테스트베드는 5대의 PC를 이용

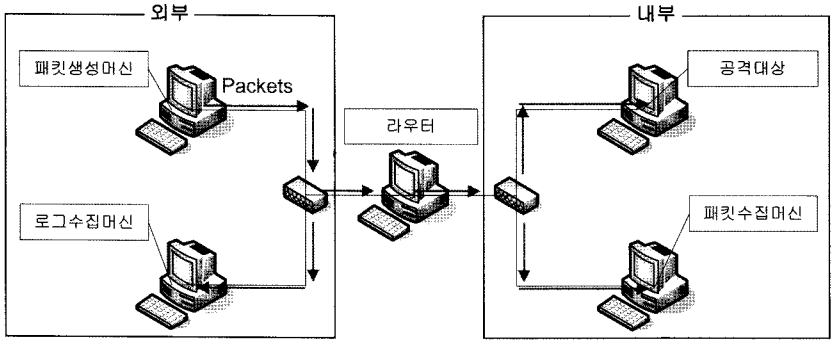

그림 2 테스트베드 하드웨어 구성

하여 패킷생성머신, 로그수집머신, 라우터, 공격대상, 패킷수집머신을 구성하였다.

외부에는 패킷생성머신과 로그수집머신이 존재한다. 패킷생성머신은 공격 도구를 이용하여 공격 트래픽을 생성하거나 데이타셀에 담긴 트래픽의 시간 정보와 패킷 정보를 이용하여 패킷 생성 시간 간격에 맞도록 패킷을 재생시키는 역할을 수행한다. 로그수집머신은 침입탐지시스템과 같은 보안 어플리케이션을 설치하여 패킷생성머신에서 생성하는 트래픽에 대한 로그를 생성하고 수집한다. 또, 트래픽을 시각적으로 표현하는 네트워크 모니터링 툴을 이용하여 네트워크 모니터링 역할도 수행한다. 로그수집머신은 promiscuous 모드로 외부네트워크에서 전송되는 모든 패킷들을 수집하고, 모니터링 툴과 침입탐지시스템으로 수집된 패킷들을 전달한다. 결과적으로 로그수집머신에서 동작하고 있는 응용프로그램들은 패킷생성머신에서 생성된 모든 패킷들을 읽을 수 있고 로그에 패킷들의 정보가 기록된다.

라우터는 외부의 패킷생성머신으로부터 생성되는 패킷들이 목적지 역할을 하는 내부로 유입되도록 설정하며 일정한 delay를 유발시켜 인터넷의 WAN환경을 제공한다. 또, 방화벽을 설치하여 내부로 유입되는 트래픽에 대한 로그를 생성하고 내부에서 외부로 유출되는 패킷을 구분하여 막는다. 외부에서 생성되어 내부로 유입되는 패킷들에 대한 기록을 라우터에 설치된 방화벽으로부터 얻을 수 있다. 또한, 라우터의 방화벽은 내부 네트워크에서 외부 네트워크로 전달되는 모든 패킷들을 차단한다. 공격 패킷을 받은 공격대상은 자체의 프로토콜에 따라 외부 네트워크로 응답 패킷들을 보내는데, 만일 이 응답 패킷들이 외부로 전달이 되면 이미 데이타셀에 포함된 응답 패킷들을 포함해 두종류의 응답 패킷들을 로그기록머신이 보게되어 원치 않은 내용이 로그에 남게 되기 때문이다.

내부에는 공격대상 및 패킷수집머신이 존재한다. 공격대상은 생성되는 공격의 목적지에 해당하는 머신으로, 공격 시 공격 대상에게 나타나는 로그를 수집하기 위해 사용된다. 데이타셀으로부터 공격을 재생할 경우, 데이타셀에 기록되어 있는 실제 공격 대상의 IP주소와 동일하게 설정해야 한다. 패킷수집머신은 공격 대상을 제외한 목적지로 향하는 패킷을 흡수하는 역할을 수행한다. 패킷수집머신은 패킷생성머신에서 생성된 패킷이 라우터를 거쳐 내부로 전송될 수 있도록 패킷의 목적지 역할을 한다.

외부 네트워크의 패킷을 내부 네트워크로 라우팅하기 위해 라우터는 내부 네트워크과 연결된 인터페이스를 통해 목적지 IP 주소의 MAC 주소를 묻는 ARP 패킷을 보낸다. 실제 네트워크에서는 목적지 IP 주소의 MAC 주소나 또는 default router 가 ARP 패킷의 응답으로 돌아오지만 본 테스트베드에서는 내부 네트워크의 패킷수집머신이 모든 목적지 IP 주소를 대신해서 임의로 ARP 응답 패킷을 만들어 전송한다. 만일 패킷수집머신이 ARP 응답을 하지 않는다면 라우터는 목적지 IP 주소가 존재하지 않는 것으로 인식하고 외부 네크워크에 ICMP destination unreachable 패킷을 보낼 것이다. 그렇게 되면, ICMP 에러 메세지를 외부 네트워크의 로그기록머신이 읽게 되고 원치 않은 내용이 로그에 남게 된다. 내부 네트워크에 있는 공격대상은 미리 데이타셀에 있는 공격대상의 IP 주소로 설정을 해 놓았으므로, 공격대상으로 향하는 패킷은 패킷수집머신의 방해를 받지 않고 공격대상으로 전달이 된다. 그 외의 패킷들은 내부네트워크에서 더 이상 처리되지 않고 없어지게 된다.

표 1은 테스트베드를 구성하는 5대 PC의 하드웨어 구성을 나타낸다. 사용한 5대의 장비는 Linux와 FreeBSD, WindowsXP 운영체체를 사용하였으며, Intel Pentium 3, Intel Pentium 4 CPU를 각각 사용하였다. 패킷생성머신은 트래픽 생성에 사용되는 데이타셀을 저장하기 위해서, 로그수집머신은 생성되는 로그를 저장하기 위해 타 장비에 비해 고용량의 HDD를 사용하였다.

3.2 소프트웨어 구성요소

테스트베드는 앞서 살펴본 바와 같이 외부 네트워크의 패킷생성머신 및 로그수집머신, 내부 네트워크의 공격대상 및 패킷수집머신으로 구성되어 있으며 두 네트워크 사이는 라우터로 연결되어 있다. 그림 3은 테스트베드에 설치된 소프트웨어의 구성을 보여준다.

외부 네트워크의 패킷생성머신에는 데이타셀을 이용하여 트래픽을 재생시킬 수 있는 tcpreplay[6]를 설치하였다. Tcpreplay는 패킷들이 pcap 파일 포멧으로 저장

표 1 하드웨어 구성정보

H/W	하드웨어 구성정보			운영체제		
	CPU	RAM	HDD	Fedora	FreeBSD	XP
패킷생성머신	Intel Pentium-4 1.7GHz	512MB	200GB	O	X	X
로그수집머신	Intel Pentium-4 1.6GHz	512MB	120GB	O	O	X
라우터	Intel Pentium-3 933MHz	256MB	40GB	O	X	X
공격대상	Intel Pentium-3 800MHz	256MB	40GB	O	O	O
패킷수집머신	Intel Pentium-3 800MHz	256MB	20GB	O	X	X

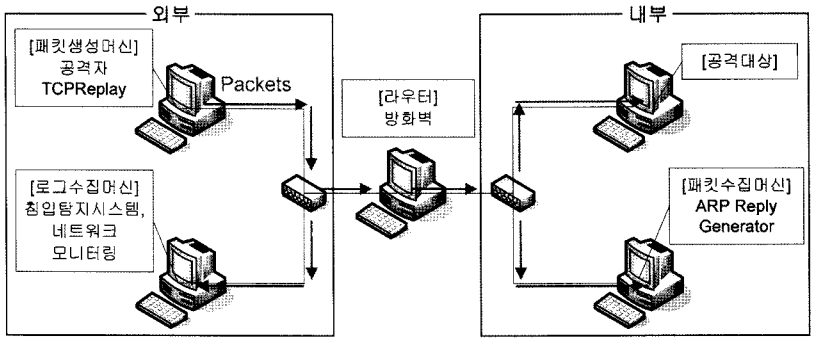

그림 3 테스트베드 소프트웨어 구성

된 데이타셑을 이 용하여 패킷을 재생하는 어플리케이션이다. Pcap 헤더에는 데이타셑이 수집되었을 당시의 timestamp가 microsecond 단위로 저장되어 있어, 데이타셑 생성 시 기록된 패킷들의 시간 간격과 동일한 간격으로 패킷을 재생할 수 있다. Tcpreplay는 timestamp와 패킷의 헤더 및 payload를 이용하여 패킷을 재생하며, 패킷의 재생속도 조절 및 패킷 목적지의 MAC주소를 변경할 수 있는 옵션이 제공된다.

로그수집머신에는 로그 수집과 공격 탐지 및 트래픽 모니터링을 위해 침입탐지시스템인 bro[7], snort[8]와 네트워크 모니터링 어플리케이션인 ntop[9], etherape[10]을 설치하였다.

Bro는 비정상행위 기반 침입탐지시스템이다. 네트워크에 나타나는 비정상적인 트래픽에 대한 경고를 하며 탐지하고자 하는 비정상행위에 따라 제공되는 policy를 선택할 수 있도록 구성되어 있다. 또, 공격 탐지를 위해 공격 시그니처도 등록하여 사용할 수 있으며, 공격에 대한 경고와 함께 해당 경고를 발생시킨 트래픽도 pcap 파일 포멧으로 저장할 수 있다.

Snort는 시그니처 기반의 침입탐지시스템이다. 공격 패킷의 헤더와 payload로부터 얻을 수 있는 시그니처를 탐지 룰로 하여 룰에 일치하는 패킷이 발견될 경우 경고 메시지를 생성한다. 새로운 공격에 대한 룰은 snort 홈페이지를 통해 제공되며, 사용자가 직접 시그니처를 생성하여 추가할 수도 있다.

Ntop은 네트워크 모니터링을 수행하는 도구로서 네트워크에 나타나는 패킷 정보를 수집하여 호스트별 트래픽 송수신량 및 호스트별 송수신 프로토콜 비율, 네트워크에 마지막으로 패킷을 송수한 시간 정보들을 제공한다. 호스트 별로 수집한 정보를 이용하여 해당 네트워크의 전체 트래픽 통계를 표시하며, 시간에 따른 트래픽 양의 변화와 전체 트래픽에 대한 프로토콜의 비율도 제공한다.

Etherape은 네트워크에 전송되는 패킷들의 전송과정 및 전송량을 비주얼하게 표현해 주는 소프트웨어로, 프로토콜 유형을 구분하고 패킷 전송량을 표현할 수 있다. 따라서 하나의 목표로 많은 공격 패킷이 집중되는 DDoS 공격과 같이 시각적으로 확인할 수 있는 정보들을 제공해준다.

라우터에는 로그 수집을 위해 방화벽인 iptables[11]를 설치하였다. Iptables는 설정된 룰에 따라 패킷을 차단하거나 패킷에 대한 로그를 생성하는 방화벽이다. 구축한 테스트베드에서는 라우터 머신에 설치하였으며 외부에서 내부로 전송되는 패킷에 대한 로그를 생성하는 역할을 한다. 패킷 필터링 룰을 정의하여 패킷을 차단할 수 있으나 테스트베드에서는 모든 패킷을 통과시키고 로그를 수집하기 위해 사용되었다.

내부에는 공격대상이외의 목적지로 향하는 패킷들이 정상적으로 라우터를 거쳐 내부로 전송될 수 있도록 패킷생성머신을 자체 개발[17]하여 패킷생성머신에 설치하였다.

표 2는 테스트베드를 구성하는 소프트웨어를 나타내는데 사용목적과 설치한 어플리케이션의 버전을 나타낸다.

표 2 소프트웨어 구성정보

S/W	기능	테스트베드에서의 사용 목적	버전
tcpreplay	트래픽 재생	데이타셑을 이용하여 트래픽 재생	2.3.5
ntop	네트워크 모니터링	트래픽 로그 수집	3.2rc2
bro	공격 탐지 및 경고	침입탐지시스템 (로그 수집 및 공격 탐지)	0.9
snort	공격 탐지 및 경고	침입탐지시스템 (공격 탐지)	2.4.3
etherape	네트워크 모니터링	네트워크 모니터링	0.9.1
iptables	정의된 패킷 차단	Firewall (트래픽 로그 수집)	1.3.0

3.3 데이타셋

다양한 공격이 포함되어 있는 데이타셋을 수집하는 것은 테스트베드 구축에 있어서 가장 중요한 일 중 하나이다. 인터넷 여러 곳에서 여러 형태의 네트워크 공격이 일어나고 있지만 유용한 데이타셋을 구하기는 쉽지 않다. 이것은 첫째로 사설 네트워크에 접근해 트래픽을 파일에 저장하기가 쉽지 않기 때문이고, 둘째는 ISP (Internet Service Provider) 등이 트래픽 분석을 위해 트래픽을 파일로 저장하곤 하지만 가입자 개인의 프라이버시 보호를 이유로 공개하지 않기 때문이다. 데이타셋을 공개할 경우에는 포함된 개인정보나 기밀사항이 담긴 패킷의 payload를 삭제하거나 IP주소를 임의의 주소로 변경하기 때문에, 인터넷을 통해 수집할 수 있는 데이타셋들 역시 불완전한 패킷들로 구성된 경우가 많다. 그럼에도 불구하고, 우리는 인터넷 다섯 곳에서 실제공격이 포함되어 있는 데이타셋을 수집할 수 있었다. 수집한 데이타셋은 KDD Cup 1999 데이타셋[12], MIT Lincoln Lab 데이타셋[13], NLANR 데이타셋[14], CAIDA[15], SONY MAWI 데이타셋[16]이 있다.

표 3은 수집한 데이타셋에 내재된 공격과 패킷수집크기, 사용목적과 수집시기, 평균용량, 파일포멧을 나타낸다. 본 논문에서는 실제 공격과 유사한 트래픽을 재생하기 위해 인터넷에 공개된 여러 데이타셋을 수집하여 분석하였고, 포멧을 변환하거나 payload가 일부 삭제된 패킷을 수정하여 테스트에 사용가능 하도록 하였다.

데이타셋은 pcap 파일 포멧이 표준이지만 다른 포멧 역시 사용이 가능하다. 수집한 데이타셋 중 NLANR 데이타셋의 경우 tsh 포멧을 사용하기 때문에 tcpreplay를 이용하여 공격을 재생하기 위해서는 pcap 파일 포멧으로 변환해야 한다. 따라서 우리는 tsh 포멧을 pcap 포멧으로 변환하는 프로그램을 구현하였다[18]. Tsh 포멧은 pcap 포멧에 비해 패킷의 헤더가 약 4Bytes 짧게 기록되어 있으며 패킷의 payload는 기록되어 있지 않다. Tsh 포멧을 pcap 포멧으로 변환하기 위해서는 tsh 포멧에 기록된 패킷의 헤더정보를 pcap 포멧에 맞게 재배치하고 tsh 포멧에는 포함되어 있지 않은 TCP checksum필드와 Urgent pointer필드를 추가해주어야 한다.

SONY MAWI 데이타셋은 앞에서 최대 96Bytes까지 저장되어 있다. 실제 데이터셋에 저장된 패킷 길이는 원래의 패킷 길이와 관계없이 96Bytes인 반면, 헤더에는 원래의 패킷 길이가 저장되어 있고 checksum 역시 원래 패킷에 해당하는 checksum 정보가 저장되어 있다. 이 정보들을 수정하지 않고 데이타셋을 재생시키게 되면 패킷 헤더에 저장된 정보와 실제 재생된 패킷이 불일치하게 되어 패킷이 폐기된다. 따라서 정상적으로 패킷을 재생하기 위해서는 패킷 길이, checksum 등 패킷 헤더에 기록된 패킷 정보들을 반드시 수정해야 한다. 따라서 우리는 slammer 웜에 해당하는 패킷에 payload를 복원하는 프로그램을 개발하여 실제 웜 패킷이 재현되도록 데이타셋을 가공하였다[19]. 본 연구에서는 slammer 웜의 재현을 위해 이 데이타셋을 사용하였다.

3.4 수집 가능한 로그

공격 발생시에 나타나는 공격의 징후정보는 네트워크에 존재하는 다양한 센서들의 로그를 이용하여 얻을 수 있다. 이 징후정보에는 네트워크 트래픽의 흔적이나 호스트에 대한 접근기록 또는 침입탐지시스템과 같은 방어 시스템에 남겨지는 스캐닝, 공격 과정에서 발생하는 비정상적인 트래픽의 증가 기록 등이 있다[21]. 본 테스트베드는 인터넷에 존재하는 센서들을 직접 배치하여 다양한 로그를 수집할 수 있도록 구성되었다. 이러한 로그는 로그수집머신에 설치된 모니터링 툴과 침입탐지시스템으로부터 얻을 수 있으며 라우터에 설치한 방화벽 로그와 공격대상에 기록되는 호스트 로그 역시 수집이 가능하다.

표 4는 로그별로 포함된 주요 정보들을 나타낸다. 네트워크 모니터링 툴인 ntop은 웹브라우저를 통해서 로그를 수집할 수 있다. 로그 파일은 사용자의 편의에 따라 PHP(Personal Hypertext Preprocessor), PERL(Practical Extraction and Report Language), Python, Text의 4가지 형식으로 저장이 가능하다. DDoS 공격과 같이 특정 호스트에 공격이 집중되어 나타나는 공격에 대한 공격대상을 추정하는 정보로 활용하기에 좋다.

비정상행위 기반 침입탐지시스템인 bro는 스캐닝과 같이 시그니처 기반의 침입탐지시스템으로 탐지가 어려

표 3 데이타셋 정보 및 용도

데이타셋	내재된 공격	패킷수집크기	사용목적	수집시기(년)	용량(MB)	포멧
KDD Cup 1999	DoS, Backdoor, Buffer overflow	헤더길이	침입탐지시스템 성능평가	1999	17	Pcap
MIT lincoln lab	DoS, DDoS	제한없음	침입탐지시스템 성능평가	1998, 1999, 2000	150~200	Pcap
NLANR	Slammer, Codered	헤더길이	트래픽수집, 분석	2001	25~40	Tsh
CAIDA	Witty	없음	트래픽수집, 분석	2001	1	Text
SONY MAWI	Slammer, Witty	96 bytes	트래픽수집, 분석	1999~현재	100~150	Pcap

표 4 로그에 포함된 주요정보

로그	A	B	C	D	E	F	G	H	I	J	K	L
Ntop	O	O	O	X	X	X	O	X	O	X	O	O
Bro	O	O	O	O	X	O	X	O	X	O	X	X
Snort	O	O	O	O	O	O	X	O	X	O	X	X
Iptables	O	O	O	O	O	X	X	X	X	X	X	X
Windows	O	O	O	O	O	X	X	X	X	X	X	X
FreeBSD	O	X	X	X	X	X	O	X	X	X	X	X

(A:time stamp, B:IP주소, C:Port번호, D:패킷길이, E:Payload정보, F:공격유형, G:수신 총 패킷수, H:수신 공격 패킷수, I:송신 총 패킷수, J:송신 공격 패킷수, K:수신 byte수, L:송신 byte수)

운 공격에 대한 탐지를 위해 활용된다. 특히 공격이 나타날 때의 징후정보를 얻기에 적합하다. 시그니처 기반의 침입탐지시스템인 snort는 정의된 시그니처와 일치되는 패킷을 공격으로 분류하고 로그를 생성한다.

방화벽인 iptables는 패킷 필터링 룰을 정의할 때 로그 생성여부를 포함하여 정의한다. 본 테스트베드에서는 방화벽을 거쳐가는 패킷들의 정보를 모두 로그로 기록하도록 설정하였다. ICMP unreachable 메시지와 같이 비정상적인 전송으로 인해 생성되는 패킷들에 대한 정보를 수집하기 적합하며 웜과 같이 임의의 목적지로 전파되는 공격의 탐지에 활용이 가능하다.

공격대상이 역할을 하는 컴퓨터에는 WindowsXP와 FreeBSD 운영체제를 설치하였으며, 각각의 운영체제를 부팅을 시킨 후 실험을 수행하여 호스트 로그를 생성하였다. WindowsXP의 경우, 서비스팩 2를 설치한 후 Windows 방화벽을 사용하여 호스트 로그를 생성하였다. DDoS 공격과 같이 많은 공격 패킷이 동시에 접근할 경우 해당 패킷들을 모두 차단하며 이를 기록한다. 전송된 패킷의 payload 정보도 기록되는데 TCP 패킷의 경우 TCP 헤더에 포함된 플래그값도 기록되기 때문에 TCP-syn flooding과 같은 공격에 대한 정보를 얻기에 적합 하다. FreeBSD의 경우, WindowsXP 만큼 상세한 정보가 기록되지는 않는다. FreeBSD는 운영체 제가 정상적으로 처리할 수 있는 패킷의 수를 초과할 경우 이에 대한 경고와 도착한 패킷의 수를 명시하여 로그로 제공한다. 호스트 로그의 수집을 위해 Linux도 설치하여 테스트하였으나 Linux운영체제에서는 네트워크 공격에 대한 어떠한 로그도 얻지 못하여 호스트 로그 수집에 사용하지 않았다.

4. 공격의 재현

본 장에서는 제안한 테스트베드와 수집한 데이타셑을 이용하여 DDoS 공격과 웜을 재현한다. DDoS 공격은 하나의 공격 대상을 향해 공격 패킷들이 집중되는 형태를 띠며, 웜은 불특정 다수로 감염 패킷이 확산되는 공격 형태를 띤다. 각각의 공격 양상에 맞도록 테스트베드를 설정한 후 공격을 재현하였고, 침입탐지시스템과 네트워크 모니터링 어플리케이션을 이용하여 공격이 정상적으로 재현됨을 확인하였다.

4.1 DDoS 공격의 재현

수집한 데이타셑들 중에서 DDoS 공격의 재현에 사용된 데이타셑은 MIT Lincoln Lab의 2000년 데이타셑이다. MIT Lincoln Lab 2000년 데이타셑에는 IP주소가 131.84.1.31인 머신을 공격 대상으로 한 DDoS 공격이 포함되어 있으며 패킷의 payload가 모두 기록되어 DDoS 공격 재현에 적합하였다.

그림 4는 DDoS 공격의 재현을 위한 테스트베드의 구성을 나타낸다. 테스트베드의 외부를 구성하는 패킷생

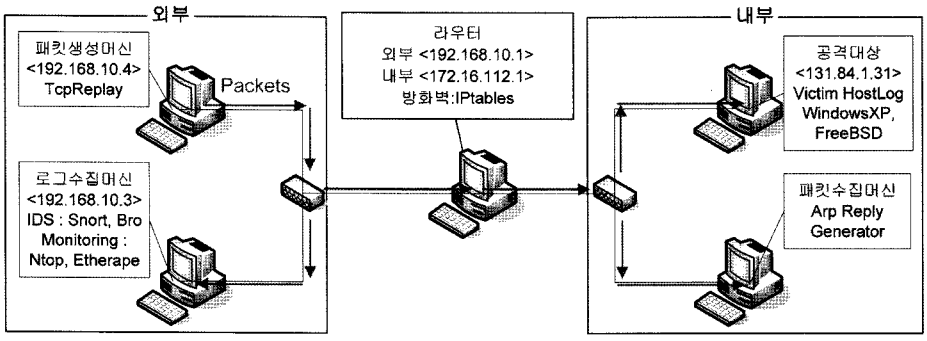

그림 4 DDoS 공격의 재현을 위한 테스트베드의 구성

성머신과 로그수집머신은 임의의 IP주소를 할당하였다. 외부를 구성하는 두 머신은 DDoS 공격이 포함된 데이타셀을 이용하여 패킷을 재생하고 로그를 수집하는 역할만 수행하므로 임의의 IP주소를 사용하여도 무방하다.

공격대상은 파티션을 나누어 WindowsXP와 FreeBSD를 설치하였다. 호스트 로그의 수집을 위해 각 운영체제를 한 번씩 구동하여 총 두 번에 걸쳐 동일한 실험을 수행하였다. 공격대상의 IP주소는 MIT Lincoln Lab 2000년 데이타셀에 포함된 공격 정보를 이용하여 공격대상의 IP주소인 131.84.1.31으로 설정하였다.

수집된 로그는 ntop, bro, iptables, WindowsXP 호스트로그, FreeBSD 호스트로그이며 snort 로그는 수집되지 않았다.

4.2 웜의 재현

테스트베드를 이용한 웜의 재현을 위해 slammer 웜을 포함하고 있는 SONY WIDE MAWI Working Group에서 제공하는 데이타셀을 이용하였다. SONY WIDE MAWI Working Group의 데이타셀 외에 NLANR에서 제공하는 데이타셀에도 역시 slammer 웜과 codered 웜이 포함되어 있었으나, 웜을 재현하기에는 부적합하였다. Tcpreplay를 이용해 NLANR의 데이타셀을 재생하기 위해 우리는 자체 개발한 프로그램[18]을 통해 tsh 포멧의 파일을 pcap 포멧으로 변환하였으나, IP 주소가 임의의 주소로 변경되어 있고 payload가 기록되어 있지 않았다. 한편 SONY WIDE MAWI의 데이타셀은 웜 패킷이 재현되도록 3.3과 같이 데이타셀을 가공하여 본 연구에 이용하였다.

그림 5는 slammer 웜의 재현을 위한 테스트베드의 구성을 나타낸다. 테스트베드 외부는 DDoS 공격 재현의 경우와 동일하게 구성하였다. DDoS 공격 재현과 차이점은 테스트베드 내부의 공격대상 머신을 사용하지 않았다는 것이다. Slammer 웜의 공격의 재현은 웜의 감염 패킷이 확산되어 나가는 과정을 관찰하는데 초점을 맞추었기 때문에, 공격의 대상이 되는 공격대상은 사용하지 않았다.

수집된 로그는 ntop, bro, snort, iptables 로그이다. 특히 iptbales 로그에 기록된 ICMP unreachable 메시지를 이용하여 웜이 임의의 목적지로 전파됨을 확인할 수 있었다.

5. 테스트베드 구성상의 문제 및 해결

본 테스트베드를 구축하는 과정에 발생했던 하드웨어 구성상의 문제, 소프트웨어 설치 및 사용상의 문제, 데이타셀 수집 및 사용상의 문제점과 해결방법을 소개한다.

5.1 하드웨어 구성상의 문제

하드웨어 구성상의 문제는 크게 패킷 포워딩 및 side effect 문제, 재생되는 트래픽의 전송문제가 있다.

첫째, 패킷 포워딩 및 side Effect 문제는 라우터의 패킷 포워딩 처리와 데이타셀을 이용하여 공격을 재현할 경우 발생하는 중복된 응답 패킷 처리 문제를 뜻한다. 데이타셀으로부터 재생되는 패킷에는 공격 패킷과 함께 공격에 대한 응답 패킷이 포함되어 있다. 따라서 데이타셀에 포함된 공격에 대한 응답 패킷과 테스트베드 내부에 존재하는 공격대상 머신이 공격 패킷을 받았을 때 생성되는 응답 패킷은 동일한 공격에 대한 중복된 응답이다. 이에 대한 해결책은 라우터에 있는 iptblaes를 이용하여 공격대상으로부터 발생되는 응답 패킷이 외부네트워크에 있는 로그수집머신으로 전달되는 것을 막는 것이다.

둘째, 인터넷상에서 수집한 데이타셀에 존재하는 무수한 목적지 IP 주소들은 본 테스트베드에서는 존재하지 않는 주소들이다. 이렇게 테스트베드에서 존재하지 않는 목적지 IP를 가지 패킷들이 재생되면 외부네트워크에서 내부네트워크로 라우팅이 되지 않는다. 그 이유는 라우터가 내부네트워크에서 목적지 IP 주소로 ARP 메세지를 보내지만 실제로 내부네트워크에서 목적지 IP 주소

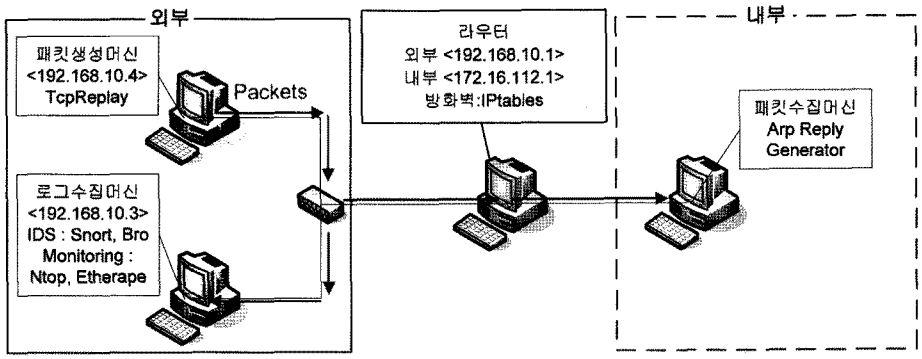

그림 5 Slammer 웜의 재현을 위한 테스트베드의 구성

도 default gateway 도 존재하지 않기 때문이다. 따라서 테스트베드의 외부네트워크에서 내부네트워크로 패킷을 라우팅하기 위해서는 내부네트워크에 해당 목적지 IP 주소가 존재하는 것처럼 ARP 메세지에 응답을 보내주면 된다. 우리는 내부네트워크 한 컴퓨터에서 packet absorber 라는 ARP 응답 대몬을 제작하여 내부네트워크의 모든 ARP 요청 메세지에 응답하게 했다. 그 결과, 외부네트웨크에서 내부네트워크로의 원활한 라우팅이 이루어질 수 있었다.

5.2 소프트웨어 설치 및 사용상의 문제

소프트웨어 설치 및 사용상의 문제점으로는 tcpreplay를 이용한 패킷 재생문제, bro설치 문제, virtual host 생성문제, 공격대상의 호스트로그 생성 문제가 있다.

첫째, tcpreplay를 이용한 패킷 재생 시 패킷들이 default gateway로 전송되지 않아 발생하는 문제이다. 정상적인 경우 패킷의 목적지 IP 주소가 같은 서브넷에 존재하는 주소이면 바로 전달이 되지만 외부 주소이면 default gateway로 패킷을 전송하고 3계층의 라우팅을 통해 해당 목적지 IP 주소가 있는 네트워크로 패킷을 전송한다. 외부네트워크에서 재생되는 패킷들은 라우터 즉, default gateway를 통해 모두 내부네트워크로 전송이 되도록 본 테스트베드에서는 설계되어 있다. 그러나, 데이타셀을 통해 재생되는 패킷들의 목적지 IP 주소와 MAC 주소들은 저장당시에 유효한 주소들을 가지고 있기때문에 목적지 주소를 라우터의 MAC 주소로 변환을 해주어야 설계된 대로 테스트베드가 동작하게 된다. 데이타셀 재생 프로그램인 tcpreplay는 -D 옵션을 통해 지정된 MAC주소로 패킷이 전송되도록 하는 기능을 제공한다. 패킷 전송 문제는 이 tcpreplay의 -D 옵션을 이용하여 테스트베드의 외부와 연결된 라우터의 네트워크 인터페이스 MAC주소로 패킷을 전송함으로써 해결되었다.

둘째, bro 설치의 문제는 Linux에서 bro가 정상적으로 동작하지 않는 문제를 뜻한다. Bro는 FreeBSD와 Linux에 설치가 가능한 비정상행위 기반의 침입탐지시스템이지만 Linux(Fedora Core 4)에 설치하여 사용을 시도한 결과 정상적으로 동작하지 않는 문제가 발생하였다. Bro의 데몬을 실행시키기게 되면 실행 시 정의된 Policy 설정에 따라 비정상 트래픽에 대한 로그를 생성해야 정상적으로 동작하는 것인데, Linux에서는 policy에 관계없이 bro 데몬이 에러를 발생하며 실행되지 않거나 실행되더라도 아무런 로그를 생성하지 않는 문제를 보였다. 반면 FreeBSD에 bro를 설치한 경우 정상적으로 동작하는 것을 확인하였다. 테스트베드에서는 bro의 경우 FreeBSD에 설치하여 사용하였다.

셋째, 테스트베드를 이용한 실험에서 여러대의 호스트가 요구될 때 실제로 여러대의 장비를 사용하여 테스트베드를 구성하면 많은 비용이 소모되고 관리면에서도 어려움을 겪게된다. 여러대의 호스트를 적은 수의 장비를 사용하여 구현할 수 있는 방법 중 하나가 가상머신(virtual host)이다. 가상머신을 이용하면 테스트베드를 구성하면 한대의 컴퓨터를 이용하더라도 마치 네트워크 상에서 여러대의 호스트가 존재하는 것과 같은 효과를 얻을 수 있다. 가상머신을 설정하기 위해서는 물리적으로 하나의 네트워크 인터페이스에 여러 개의 이름을 생성하고 각 이름들에게 서로 다른 IP주소를 할당하는 방법을 사용한다. 예를 들어 eth0와 같은 인터페이스는 ifcfg-eth0:0, ifcfg-eth0:1 등과 같이 여러개의 인터페이스 이름을 붙일 수가 있고 각각이 이름에 IP 주소를 부여할 수 있다. 한개의 물리적인 인터페이스에는 최대 255개의 IP 주소를 부여할 수 있다. 이렇게해서 만들어진 복수개의 인터페이스 이름은 커널상에서는 공통의 TCP/IP 스택을 공유하게 된다. 따라서 IP 주소의 숫자가 많아지면 컴퓨터의 성능에 역으로 영향을 미칠 수 있다.

넷째, 호스트 로그 수집문제는 공격대상이 되는 호스트 머신의 시스템 로그를 얻는데 발생하는 문제점을 뜻한다. DDoS와 같은 공격의 공격 대상에게 남겨지는 호스트 로그를 얻기 위해 다양한 운영체제를 이용하여 로그를 생성시켜 본 결과, Linux(Fedora Core 4)의 호스트 보다는 FreeBSD의 호스트에서 보다 자세한 로그를 얻을 수 있었다. 예를 들어, Linux의 경우 DDoS 공격에 대해 로그가 제한적으로 생성이 된 반면, FreeBSD의 경우 초당 200개가 넘는 패킷이 전송되었다는 경고 메시지가 생성되었다. Windows의 경우 Windows 내에 방화벽을 사용하여 로그를 수집하였다. Windows 로그에는 전송되어 온 모든 패킷에 대한 정보와 해당 패킷에 대한 응답정보가 기록되었다.

6. 결 론

보안 제품의 성능평가나 검증을 위해서는 실제 네트워크에서 공격이 발생하는 경우에 대한 테스트가 요구되며, 이를 위해 실제 네트워크에서 공격이 발생하는 것과 유사한 테스트 환경을 제공해 줄 수 있는 네트워크를 구축하여 테스트를 진행하여야 한다. 본 논문에서는 적은 수의 PC로 실제 인터넷과 유사한 환경을 구축할 수 있는 테스트베드를 제안하였다. 제안하는 테스트베드는 테스트베드의 사용 목적이나 공격의 유형에 따라 수정이나 확장이 용이하도록 하였으며, 공격을 직접 재생하거나 데이타셀을 이용하여 재현할 수 있는 환경을 제공하였다. 테스트를 위해 공격 트래픽을 포함하는 여러 데이타셀들을 인터넷을 통해 수집하였고 각 데이타셀에

포함된 패킷을 재현하기 위해 포멧을 변환하거나 패킷 헤더 정보를 수정하여 사용하였다. 또, 테스트베드를 구축하는 과정에 나타나는 하드웨어 구성의 문제, 소프트웨어 설치 및 사용상의 문제, 데이타셋 수집 및 사용상의 문제를 설명하고 이에 대한 해결법을 제시하였다. 구축한 테스트베드와 수집한 데이타셋을 이용하여 실제 인터넷에서 발생하는 것과 유사한 DDoS공격 및 웜을 재생해 보였으며 침입탐지시스템, 침입차단시스템, 네트워크 모니터링 도구를 이용하여 보안장비의 성능평가 및 네트워크에 존재하는 센서들의 로그 및 공격 대상의 로그도 수집하였다. 본 테스트베드는 많은 비용과 시간을 투자하지 않고서도 네트워크에 존재하는 센서들의 로그 수집 또는 보안 장비의 성능평가가 가능하도록 구성되어 있다. 구축한 테스트베드를 이용하여 알려진 공격이나 알려지지 않은 공격을 직접 재생하고 공격 유형의 모델을 정립하거나 보안장비의 성능평가 및 다양한 공격에 대한 데이타셋 생성을 진행할 계획이다.

참 고 문 헌

[1] Dean Turner et al., "Symantec Internet Security Threat Report Trends for July 05 - December 05 Volume IX, March 2006," Symantec, March 2006.

[2] Richard P. Lippmann et al., "Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation," *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, vol.2, pp.12-26, January 2000.

[3] Cyber Defense Technology Experimental Research project, "DETER," available at http://www.isi.edu/deter/.

[4] Gautam Singaraju, Lawrence Teo and Yuliang Zheng, "A Testbed for Quantitative Assessment of Intrusion Detection Systems using Fuzzy Logic," *Second IEEE International Information Assurance Workshop (IWIA'04)*, pp.79-83, January 2004.

[5] Jelena Mirkovic et al., "Measuring Denial of Service," Quality of Protection Workshop co-located with CCS 2006, October 2006.

[6] Edgewall Software, "Tcpreplay," available at http://tcpreplay.synfin.net/trac/

[7] Lawrence Berkeley National Laboratory, "Bro Intrusion Detection System," available at http://www.bro-ids.org

[8] Sourcefire, "Snort," available at http://www.snort.org

[9] ntop.org, "Ntop," available at http://www.ntop.org

[10] Juan Toledo and Riccardo Ghetta, "EtherApe," available at http://etherape.sourceforge.net

[11] The netfilter, "netfilter/iptables project," available at http://www.netfilter.org

[12] The UCI KDD Archive, "KDD Cup 1999 Data," available at http://www.ics.uci.edu/~kdd/databases/kddcup99/kddcup99.html

[13] Lincoln Laboratory Massachusetts Institute of Technology, "MIT Lincoln Laboratory - DARPA Intrusion Detection Evaluation Data Sets," available at http://www.ll.mit.edu/IST/ideval/data/data_index.html

[14] NLANR Measurement and Network Analysis Group, "NLANR PMA," available at http://pma.nlanr.net

[15] Cooperative Association for Internet Data Analysis, "Cooperative Association for Internet Data Analysis (CAIDA)," available at http://www.caida.org

[16] MAWI Working Group, "MAWI Working Group Traffic Archive," available at http://trace r.csl.sony.co.jp/mawi/

[17] HIT Testbed, "ARP_GENERATOR," available at http://hit.skku.edu/ARP_GENERATOR/

[18] HIT Testbed, "TSH2TCPDUMP," available at http://hit.skku.edu/TSH2TCPDUMP/

[19] HIT Testbed, "RESIZE_PACKET," available at http://hit.skku.edu/RESIZE_PACKET/

[20] Projects that have actively used isi.deterlab.net (Vers: 4.37 Build: 04/13/2006), "deterlab," available at http://www.isi.deterlab.net/projectlist.php3

[21] Cristina Abad et al, "Log Correlation for Intrusion Detection: A Proof of Concept," Computer Security Applications Conference, December 2003.

임 이 진

2008년 성균관대학교 한문학과(문학사) 2008년~현재 성균관대학교 전자전기컴퓨터공학과 석사과정. 관심분야는 네트워크 보안

최 형 기

1992년 성균관대학교 전자공학과(공학사) 1996년 Polytechnique University 전기전자(공학석사). 2001년 Georgia Institute of Technology 전기전자(공학박사) 2001년~2004년 미국 Lancope. Inc. 연구원. 2004년~2006년 성균관대학교 정보통신공학부 전임강사. 2006년~현재 성균관대학교 정보통신공학부 조교수. 관심분야는 인터넷 보안, 모바일 커뮤니케이션

김 기 윤

2005년 성균관대학교 정보통신공학부(공학사). 2007년 성균관대학교 정보통신공학부 컴퓨터공학과(공학석사). 2007년~현재 ㈜파이오링크 근무중. 관심분야는 인터넷 보안, 홈네트워크, 웹보안