패킷 필터링 보안 정책을 테스트하기 위한 테스트 베드 구축

국승학[0] 김현수
충남대학교 전기정보통신공학부 컴퓨터 전공
{triple888[0], hskim401}@cnu.ac.kr

CONSTRUCTION OF A TESTBED FOR TESTING SECURITY POLICIES IN PACKET FINTERING FUNCTION

Seung-hak Kuk[0], Hyeon Soo Kim
Dept. of Computer Science and Engineering, Chungnam National University

요 약

패킷 필터링은 잠재적으로 악의 있는 네트워크 패킷을 필터링하는 것이다. 패킷 필터링의 기능을 테스트하기 위해서 우리는 보안 시스템에 설정된 보안 정책이 의도한 대로 수행되는지 검증해야 한다. 그러나 기존에 이러한 기능을 테스트하기 위한 도구가 거의 없으며, 존재하는 도구는 테스트의 수행 시 테스트 케이스 선정과 테스트 결과의 판단에 있어 많은 사용자의 판단을 요구한다. 대부분의 보안 시스템 운영자는 새로운 보안 정책을 설립할 때 이를 테스트하는데 많은 부담감을 갖는다. 이에 본 논문에서는 사용자의 판단을 최소화 할 수 있는 새로운 테스트 베드를 제안하고 구현한다. 본 논문의 테스트 베드는 테스트 케이스와 테스트 오라클을 자동으로 생성한다. 그리고 생성된 테스트 오라클을 기반으로 테스트 결과를 사용자의 참여 없이 자동으로 판단한다.

1. 서 론

네트워크가 개방되고 인터넷이 급속히 성장함에 따라 정보 보안 시스템은 갈수록 중요하게 되었다. 외부 침입에 대비한 효율적인 정보 보안 시스템을 구축 운영하려면 우선적으로 선행되어야 할 것이 조직에 맞는 정보 보안 정책 수립과 정당한 정책 수행이다. 이러한 정책의 수립이 올바르게 되었는지 혹은 보안 시스템이 정상적으로 작동하는 지에 대한 테스트는 보안 시스템을 이용하고자 하는 조직뿐만 아니라 보안 시스템 개발자들에게도 검증 단계로써 매우 중요하다[1].

라우터(router), 방화벽(firewall)은 대부분 패킷 필터링을 기본 기능으로 가지고 있으며 개발 업체들은 기능 및 성능 향상에 대한 노력들을 기울이고 있다. 본 논문에서 제안하는 테스트베드는 패킷 필터링을 기능을 갖는 라우터나 방화벽 또는 소프트웨어를 연구 개발하면서 신뢰성 있고, 사용자가 사용하기 편리한 패킷 필터링 테스트베드를 제안하고 구현한다. 기존의 테스트 시스템들은 네트워크를 통해 보내진 패킷과 받아진 패킷의 로그를 분석하는 수준의 방법을 취하고 있고, 사용자가 테스트 시스템에 대한 많은 지식을 필요로 하는 단점이 있다. 또한 테스트 케이스의 선정에 있어서 사용자의 선택에 따라 신뢰성의 기복이 심하다. 그러나 본 논문에서 제안하는 시스템은 테스트 케이스의 선정과 테스트 오라클의 생성을 자동화함으로써, 테스트에 대한 많은 지식이 없는 사용자도 쉽고 편리하게 테스트를 수행할 수 있다.

2. 관련 연구

2.1 패킷 필터링 테스팅

패킷 필터링 테스트는 방화벽이나 라우터의 보안 정책, 패킷 필터링 규칙을 테스트하기 위한 것이다. 이는 지정된 규칙에 맞게 패킷이 정확하게 필터링 되는지, 즉 패킷 필터와 이를 이용한 보안 정책에 의해 시스템이 안전하게 보호될 수 있는지 테스트하기 위한 것이다. 그러나 기존의 많은 연구들은 이러한 패킷 필터링에 관한 연구보다는 침투 테스팅(Penetration Testing), 로그 검사(Log Review), 바이러스 탐지기(Virus Detectors)등 세션 레벨 이상의 연구가 활발히 이루어지고 있다. 하지만 라우터나 방화벽 같은 보안 시스템의 기본이 되는 패킷 필터링에 관한 테스트 연구들은 간과되어 몇몇 기능이 빈약한 테스트 도구들이 존재할 뿐이다. 기존의 패킷 필터링에 관한 테스트는 대부분 수동으로 이루어지고, 존재하는 도구 또한 많은 사용자의 판단을 요구하고 있다.

Ftester[2]는 방화벽의 보안정책을 테스트하기 위한 도구이다. Ftester는 패킷 생성기의 역할로 패킷을 생성하는 Ftest와 패킷 탐색기의 역할을 하는 Ftested, 기본적인 환경을 설정하는 파일인 'ftest.conf' 파일, Ftest와 Ftested의 로그를 비교하는데 필요한 스크립트를 담고 있는 Freport로 구성되어 있다. 그러나 도구의 사용에 있어서 매번 새로운 테스트 케이스에 대해 인위적인 설정과 스크립트 작성이 필요하며 분석 도구 또한 종단간의 로그에 대한 단순 비교를 제공할 뿐이다.

3. 시스템 개요

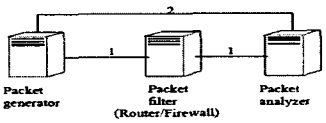


그림 1 테스트 베드

그림 1은 패킷 필터에서 패킷 필터링이 제대로 이루어지는 지 테스트하기 위해 본 시스템에서 구축한 테스트베드이며 그림 2는 그 내부 구조를 보여준다.

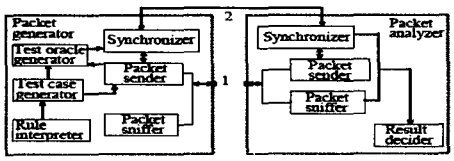


그림 2 테스트 베드 내부 구조

패킷 생성기(Packet Generator)에서는 패킷 필터에서 세운 보안 규칙을 해석하여 테스트 케이스와 오라클을 생성한 다음 테스트 케이스에 따라 패킷을 생성하여 1번 선을 통해 보내고 이에 대한 로그로 남긴다. 그 후 테스트 오라클을 2번 선을 통해 패킷 분석기에 보낸다. 패킷 분석기는 패킷 생성기에서 전송된 패킷들 중 수신된 패킷에 대한 로그를 남긴다. 다음으로 2번 선을 통해 전달된 테스트 오라클을 이용해 테스트 결과를 산출한다. 본 논문에서 제안한 테스트베드는 테스트 케이스의 생성부터 테스트 오라클의 생성 및 결과 산출까지 자동화를 할 수 있게 하였고 테스트 결과에 대해 인위적이 판단이 아닌 자동화를 실현 하였다.

4. 패킷 필터링 테스트

4.1 테스트 대상 및 방법

일반적으로 패킷 필터링 기능을 제공하는 라우터나 방화벽에서의 보안은 송신지 IP 주소, 목적지 IP 주소, 송신지 포트, 목적지 포트 및 프로토콜의 5가지 튜플의 조합으로 여러 개의 보안 정책을 세울 수 있다. 그 보안 정책은 보안의 목적과 환경에 따라 달라질 수 있으며 또한 이 5가지 튜플들과 함께 추가적인 제약사항을 기술함으로써 특별한 보안 정책을 세울 수 있는데, 이러한 특별한 보안 정책으로는 Flooding 제약 보안 정책과 상태 기반 보안 정책이 있다. 이에 본 논문에서는 테스트 대상을 다음과 같이 3가지로 분류하였다[3].

- 기본 보안 (주소, 포트, 프로토콜 기반) 정책
- Flooding 제어 보안 정책
- 상태기반 보안 정책

각각의 테스트 대상은 패킷 필터에 규칙으로 표현된다. 본 논문에서는 이러한 규칙을 해석하여 테스트에 필요한 정보를 추출하여 테스트 케이스와 오라클을 생성한다. 이때 추출되는 정보는 생성될 테스트 패킷의 기본 정보와 규칙이 표현하고 있는 허가/거부 범위이다. 본 논문에서의 테스트 방법은 이러한 허가/거부 범위를 이용해 정상적인 상황에서의 패킷을 전송하는 것과, 비정상적인 패킷을 필터링하는지 확인하는 방법을 이용한다. 다음 표1은 각 테스트 대상에 대한 테스트 방법을 보여준다.

표 1 테스트 대상별 테스트 방법

테스트 대상	정상적 상황	비정상적 상황
기본 보안 정책	주소, 포트, 프로토콜 별 허가 패킷 전송 후 정상적으로 전달되었는지 확인	주소, 포트, 프로토콜 별 거부 패킷 전송 후 정확히 필터링 되었는지 확인
Flooding 제어 정책	Non-Flooding 패킷 전송 후 모든 패킷이 전달되었는지 확인	Flooding 패킷 전송 후 제어 정책에서 허가하는 수 만큼의 패킷이 전달되었는지 확인
상태기반 정책	허가 상태 패킷 전송 후 올바른 상태 전이가 일어나는지 확인	거부 상태 패킷 전송 후 상태 전이가 일어나지 않음을 확인

4.1 테스트 케이스 및 오라클 설계

그림 3은 테스트과정에서 필요한 데이터를 추출하는 과정을 보여준다. 본 논문의 규칙 해석기는 패킷 필터에 설정된 규칙을 읽어 파싱 과정을 거쳐 생성될 패킷의 기본 정보와 테스트 대상에 대한 규칙을 추출한다.

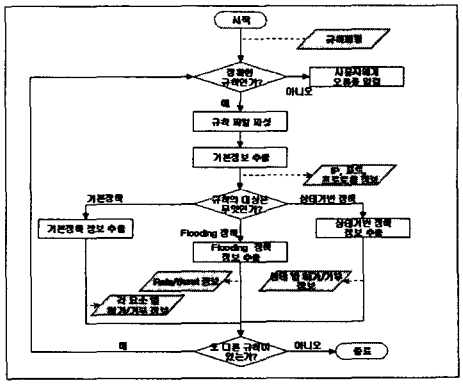


그림 3 규칙으로부터 정보 추출 과정

그림 4는 테스트 케이스 생성기와 테스트 오라클 생성기에서 테스트 케이스와 오라클을 생성하는 과정을 보여준다. 각각은 서로 다른 모듈에서 생성되지만 생성흐름이 비슷하기 때문에 하나의 그림으로 표현하였다. 각 생성 모듈에서는 규칙의 정보를 읽어 해당 규칙의 대상이 어떤 것인지 판별한다. 그리고 각 규칙이 갖는 패킷의 허가/거부 범위를 추출한다. 추출된 허가/거부 정보를 기반으로 테스트 케이스와 오라클이 생성된다.

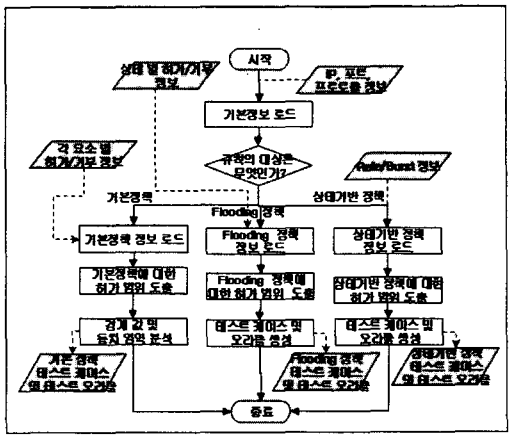


그림 4 테스트 케이스 및 오라클 생성 과정

본 논문에서는 정확한 테스트를 위해 허가/거부 정보를 바탕으로 경계값 분석 및 동치 영역 분할 방법을 이용하여 테스트 케이스를 생성한다. 아래 테스트 대상 규칙은 168.188.46.0 ~ 168.188.46.255까지의 소스 IP 주소를 갖는 패킷에 대해 허가한다는 의미를 갖는다.

테스트 대상 규칙 :-s 168.188.46.0/24 -j ACCEPT

위의 규칙은 IP 주소에 대한 규칙을 설정하는 부분으로써 먼저, IP 주소를 설정하고 Prefix에 의해 생성될 패킷의 범위를 설정하여 입력도메인으로 정한다. 이에 대한 테스트 영역은 세 개의 동치 클래스로 나뉘어 진다. 이때 생성되는 테스트 케이스는 각각의 동치 클래스로부터 추출된 IP 주소를 갖는 패킷으로 구성된다. 이는 거부 정책에 대해서도 동일하게 적

용된다. 또한 허가 범위의 경계값 부근에서 충분한 테스트 케이스를 추출한다. 즉 경계값과 경계값보다 하나 작은 값, 경계값보다 하나 큰 값의 형태로 테스트 케이스를 추출한다.

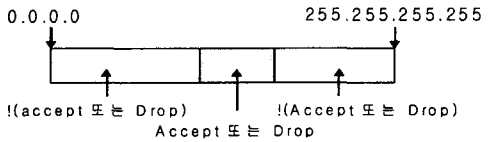


그림 5 IP주소 입력 도메인

다음 표 2는 기본 보안 정책 중 송신지 IP 주소 기반의 규칙에 대한 테스트 케이스를 보여준다.

표 2 IP 주소 기반의 규칙에 대한 테스트 케이스

TC#	Scheme	Range	Packets (IP address based)
1	Equivalence partitioning	OUT	168.188.44.74
2		IN	168.188.46.100
3		OUT	168.188.55.32
4	Boundary value analysis	OUT	168.188.45.255
5		IN	168.188.46.0
6		IN	168.188.46.255
7		OUT	168.188.47.0

다음 표 3은 기본 보안 정책에 대해 생성되는 테스트 오라클을 보여준다. 테스트 오라클은 각 테스트 케이스에 대한 테스트 수행 시 기대되는 결과 값을 나타내는 것으로 테스트 수행 결과가 옳은지 판단하는 기준으로 생각 할 수 있다.

표 3 기본 보안 정책의 테스트 오라클

Selected range	Rule	Sending packet log	Receiving packet log	Decision	Test result
IN	Accept	LogS	LogR	LogS = LogR	success
				LogS ≠ LogR	fail
OUT				LogR = Ø	success
				LogR ≠ Ø	fail
IN	Drop			LogR = Ø	success
				LogR ≠ Ø	fail
OUT				LogS = LogR	success
				LogS ≠ LogR	fail

5. 테스트 수행 및 결과

본 시스템은 생성기와 분석기 사이에 라우터를 두고 패킷 필터로 Iptables 버전 1.2.7을 이용해 보안 정책을 세우고 테스트를 수행하였다. 테스트의 수행은 본 논문에서 정의한 3가지 테스트 대상을 기준으로 수행하였으며 본 논문에서는 표2의 송신지 IP 주소 기반의 규칙을 테스트한 결과를 설명한다. 표 4는 표 2의 테스트 케이스를 기반으로 수행한 테스트를 보여준다. 여기서 선택된 테스트 케이스는 표 2의 경계 값 분석과 동치 영역 분할 방법을 이용해 최소한의 테스트 케이스를 산출한 것이다.

표 4 테스트 케이스 그룹

Test case group	Number of packets	Range(input domain)
1	10	IN
2	100	IN
3	10	OUT
4	100	OUT

본 시스템은 테스트 오라클 생성기에서 자동으로 생성되는 테스트 오라클과 테스트 수행 시 생성되는 로그 정보를 기반으로 테스트 결과를 자동으로 산출한다. 그림 5는 테스트 결과 판정기에서 테스트 결과를 산출하는 과정을 보여준다. 테스트 결과 판정기는 테스트 오라클과 수신된 패킷의 로그와 비교해 패킷을 정확히 필터링 했는지 판단한다.

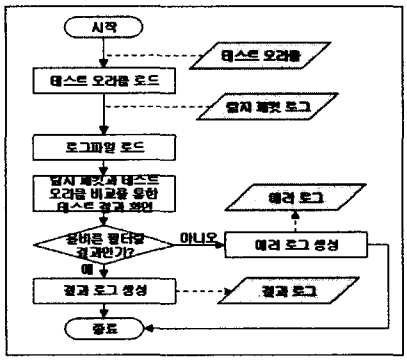
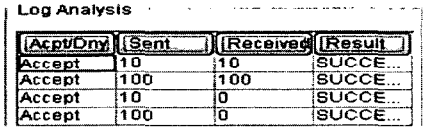


그림 6 결과 판정 과정

위 표4의 테스트 수행 결과는 그림 6에서 보여 진다. 이 결과는 본 논문에서 테스트한 라우터가 송신지 IP 주소에 관한 패킷에 대한 필터링에는 문제가 없음을 말해준다. 본 논문에서는 IP 주소 기반의 기본 정책에 대한 테스트뿐만 아니라 다른 테스트 대상에 대해서도 테스트를 수행하였으나 논문의 분량이 제한되어있기 때문에 자세한 사항은 생략한다.



Log Analysis

Acpt/Dny	Sent	Received	Result
Accept	10	10	SUCCE...
Accept	100	100	SUCCE...
Accept	10	0	SUCCE...
Accept	100	0	SUCCE...

그림 7 기본 보안 정책 테스트 결과

6. 결론 및 향후 연구방향

이번 연구를 통해 보안 시스템에 내장되어 있는 패킷 필터링 기능과 이를 이용한 보안 정책을 테스트할 수 있는 테스트 베드를 설계/구현 하였다. 테스트 베드는 기존의 테스트 도구와는 달리 테스트 케이스와 오라클의 생성을 자동화하였다. 이를 이용해 테스트의 결과 판정을 사용자의 참여 없이 자동으로 수행할 수 있다.

향후 여러 개의 규칙이 설정된 보안시스템에 대해 테스트 할 수 있도록 규칙 분석기에 대한 연구가 필요하다. 또한 이러한 규칙들 사이에서도 규칙의 순서에 의해 같은 패킷 필터링의 효과에 대해서도 시스템 성능에 영향을 미칠 수 있다. 따라서 규칙의 순서를 최적화 하는 방법에 대한 연구 또한 필요하다.

참고문헌

[1] B. Potter & G. McGraw, Software Security Testing, *IEEE SECURITY AND PRIVACY MAGAZINE*, 2(5), 2004, 81-85

[2] www.infis.univ.trieste.it/~lcars/ftester

[3] 박영대, 국승학, 김현수, 패킷 필터링을 위한 테스트베드 구축, 소프트웨어 공학회지, 2005