IP 스푼핑 탐지기술

이번 호에서는 기존 연구들의 취약점을 고려하면서 스푼핑된 패킷을 성공적으로 탐지하기 위한 방법으로 라우터에서 자거는 패킷들을 탐지하여 해당 IP 주소의 패킷이 거쳐야 하는 정상적인 경로를 지나왔는지 파악할 수 있는 새로운 탐지 기법인 BASE를 소개한다.

이러한 스푼핑 기법은 서비스 간부 공격, 세션 하이네김 등의 다른 여러 가지 공격과 함께 사용되어 네트워크에 큰 위협이 될 수 있다. IP 주소가 조작된 패킷을 차단하는 방식으로는 Ingress filtering, RPF 등이 있으나 이들은 많은 라우터에 배포되어 사용해야 효과가 있게 되는 제약이 있다. 최근 몇 년간 연구된 다른 기법들은 이를 고려하였으나 실제 적용되기에는 막대의 복잡성 및 네트워크 상황 저하 등의 문제가 있어 실제로 구현되고 적용되지 않고 있다.

IP 스푼핑(IP Spoofing)이란 IP(INTERNET Protocol) 자체의 보안 취약성을 악용한 것으로 자신의 IP 주소를 속여서 접속하는 공격을 말한다. 따라서 그림 1과 같이 신뢰 관계에 있는 두 시스템 사이에서 허가 받지 않은 자가 자신의 IP 주소를 신뢰 관계에 있는 호스트의 IP 주소로 바꾸어 속이는 것으로 rlogin, rsh와 같이 IP 주소로 인증하는 서비스를 무력화시킬 수 있다. 즉 공격자가 자신의 신뢰성을 이용한 것이므로 패킷의 소스 IP 주소를 변조하여 접속을 시도하는 접근 형태를 말한다.

그림 1. IP 스푼핑 기법을 이용한 패킷 전송

<table>
<thead>
<tr>
<th>ver</th>
<th>length</th>
<th>TOS</th>
<th>datagram length</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit identifier</td>
<td>flgs</td>
<td>fragment offset</td>
<td></td>
</tr>
<tr>
<td>time to live</td>
<td>upper layer</td>
<td>internet checksum</td>
<td></td>
</tr>
<tr>
<td>32 bit source IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 bit destination IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (if any)</td>
<td>data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(variable length, typically a TCP or UDP segment)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

그림 2. IP 패킷 헤더 구조

IP 스푼핑은 인터넷에 연결된 사용자가 IP 패킷을 마음대로 조작하여 패킷을 전송할 수 있다는 사실을 이용한다. 수신 호
스터가 패킷을 판별할 때 단지 패킷의 적합 IP 주소만을 가지고 송신 호스트를 판단하기 때문에 수신 호스트는 패킷이 어디에서 왔는지 명확하게 알 수 있는 것이 없다. 일반적인 경우 두 컴퓨터 사이의 인터넷위크가 이루어질 때 오고 가는 IP 패킷의 구조는 그림 2와 같은 구조를 따른다.

이와 같이 IP 패킷은 자신의 소스 IP 주소(32비트)와 해당 패킷이 도착해야 할 목적지 IP 주소(32비트)를 담고 있음을 볼 수 있다. 이 32비트 소스 IP 주소를 다른 신뢰한 호스트의 IP 주소로 바꾸거나 혹은 존재하지 않는 IP 주소를 이용하여 공격자가 자신의 위치를 숨기면서 탐지하기 어렵게 하는 용도로 사용된다.

주로 IP 스푸핑은 IP 주소나 호스트 이름에 의존하는 인터넷 서비스들에 대상으로 자행된다. 1995년 보안업체를 파고들어 갔던 키빈 미트닉(Kevin Mitnick)이 쓴 방법이 TCP 프로토콜의 3Way handshaking 과정을 이용한 TCP 서버와 인터넷 외부의 부분의 특정한 IP 주소를 신뢰하지 않고 할 수 있다. 이는 IP 스푸핑 공격을 대상으로 남기 일정 기간에 걸쳐 패킷들이 수신된다.

 일반적으로 TCP 프로토콜은 이용하여 호스트 사이의 통신이 이루어지기 위해서는 그림 3과 같은 세 단계의 과정을 거쳐야 한다. 그림 2에서와 같이 TCP/IP는 응답요청/응답수신 옵션협약과 같은 3단계를 거쳐서 연결이 생성된다. 다른 호스트와 통신하기 위해서 소스에서는 TCP SYN 패킷을 보낸다. 이 패킷의 목적은 서버와 통신을 담당하기 위한 것이다.

목적지 호스트는 SYN 패킷을 받았음을 알려주고 SYN/ACK 패킷을 보낸다. 마지막으로 소스 호스트는 ACK 패킷을 보낸다. 이 과정이 완료되면 두 호스트는 데이터를 교환할 준비가 끝난다. 하지만 이 과정에서 공격자는 스푸핑한 소스 IP 주소와 추측한 TCP 서버 번호를 이용해 대상 호스트를 마치 정상적인 다른 호스트와 연결된 것으로 착각하게 만들 수 있다.

실제로 이런 역할을 수행하는 것은 인터넷상에서 손쉽게 구할 수 있으므로 IP 스푸핑 공격으로부터 피해를 입지 않기 위해서는 유일하게 해결책을 최대한 빠르게 적용할 필요가 있다.

**IP 스푸핑 배경 지식**


1. **Non-Blind IP 스푸핑 공격** : 호스트 사이의 주로는 패킷을 볼 수 있는 상태에서 발생하는 것으로 시스템 번호를 알 수 있기 때문에 쉽게 공격이 가능하다.
2. **Blind IP 스푸핑 공격** : 호스트 사이의 주로는 패킷을 볼 수 없는 상태에서 시스템 번호를 알거나 정확한 주소를 알지 못해 놓아 공격자가 추정할 수 있는 공간이 줄어드는 공격이다. 하지만 대부분의 운영체제가 시스템 번호를 너무 간단한 규칙에 의하여 발생한다.

**IP 스푸핑은 보안팀의 신원을 감추기 위해 IP 패킷의 소스 주소를 변경하여 호스트에 대한 연동되지 않은 데스로 허용을 얻는데 사용된다. 인터넷 사증의 작업에서는 패킷을 전송할 목적지 IP 주소만 사용하고 소스 주소는 무시한다. 따라서 공격자는 시스템을 손상시키기 위한 패킷을 전송하고 이에 대한 소스 주소를 완전히 알 수 있기 때문에 무작정한 패킷의 출처를 알 수 없다. 스푸핑이 항상 시스템을 손상시키는 것은 아니지만 시스템에 대한 접근이 발생할 수 있음을 알려 준다. 이 주소는 접근자의 신원을 감추기 위해 사용된 외부의 주소 또는 '보안 스푸팅 공격'에 내부의 주소 또는 권한 있는 스푸팅을 가진 신뢰할 수 있는 내부 주소일 수 있다. 또한 대량의 트래픽을 발생시키는 공격의 목적 중 하나가 IP 스푸핑을 이용하는 것이다. 무한을 수행하는 패킷들 중 상당수 가 소스 IP 주소를 변조해 유입되는데 스마트(Smart) 공격, 랜드(Land) 공격, TTN(Tribe Flood Network) 공격, DoS(Dig) 공격과 같은 서비스 공격에 의한 공격을 스푸핑을 사용하는 대표적인 예이다.

**실험적인 프로토콜이 가져야 할 세 가지 특성**

"Crossing the Chasm"의 저자 제프리 무어(Geoffrey A. Moore)는 그의 저서에서 어떻게 신제품이나 새로운 아이디어 가 사람들 사이에 퍼지는데를 설명했다. 네트워크상에서의 프
로토를 사용하는 것 또한 이와 같은 경향을 보일 것이라고 생각한다. 따라서 프로토콜이 어려운 세 가지 특성을 지남으로써 실생활에 실제 적용될 수 있는 실용적인 프로토콜의 모습을 도달할 수 있다.

1. 초기 사용자에게 혜택(Initial benefit) : 실용적인 프로토콜은 초기 사용자에게 그 프로토콜을 사용할 때 혜택을 주어야 한다. 다른 사용자보다 먼저 프로토콜을 사용하는 두 사용자가 그 프로토콜의 사용에 따른 혜택을 가질 수 있도록 다른 사용자들에게 프로토콜 사용에 대한 동기 부여 역할을 수행할 수 있다. 따라서 초기 사용자에게는 프로토콜 사용에 따른 혜택 부여가 다른 사용자들이 그 프로토콜을 사용하도록 설득하는 구실을 하게 된다.

2. 사용자가 늘어날수록 점차 커지는 혜택 (Incremental benefit) : 프로토콜의 초기 사용자에게 주어진 혜택을 기반으로 점차 많은 사용자가 그 프로토콜을 이용하게 되면 그 프로토콜 사용이 사용자의 수가 늘어날수록 그 프로토콜 사용할 때 혜택 또한 점점 커지게 된다.

3. 부분적인 배포에서의 효율성(partial deployment) : 대부분의 사용자가 프로토콜을 이용하는 것이 아니라 어느 정도 이상의 사용자가 그 프로토콜을 사용하더라도 그 프로토콜의 사용에 따른 만족함만한 효과가 있다. 대부분의 사용자가 그 프로토콜을 사용하도록 하기 위해서는 많은 시간을 필요로 하기 때문에 30-50% 정도의 사용자가 그 프로토콜 사용이 충분히 효과적인 성능을 발휘할 수 있어야 한다.

기존 IP 스푹핑 탐지 기법 연구의 한계
하지만 기존의 스푹핑 탐지 기법들은 위의 이러한 세 가지의 특성을 만족시키고 있지 못하기 때문에 실생활에서 널리 사용되지 못하고 있다.

Ingress/Egress 필터링은 외부의 공격자가 내부의 호스트인 것처럼 속이는 것을 방지해 준다. 즉, 외부로부터 들어오고 외부로 나가는 패킷을 필터링 해주는 것이다. 하지만 이 필터링 기능을 사용하는 라우터는 자신을 직접적으로 보호해 주는 역할을 수행하지 못하기 때문에 널리 배포되기 어려워진다.

유니캐스트 RPF(Reverse Path Forwarding)는 라우터로 패킷이 들어왔을 때 패킷의 입력 인터페이스 상에 패킷에 적합 소스 IP 주소로의 역경로(Reverse Path Route)가 존재하는데 확인한다. 만약 패킷에 적합 소스 IP 주소가 스푹핑된 것이라면 입력 인터페이스 상에 소스 IP 주소로의 역경로가 존재하지 않는다. 즉 라우터는 역경로가 존재하는 패킷은 통과시키고 그렇지 않은 스푹핑된 패킷은 드롭한다. 이 기법 또한 이것을 사용하는 자에게 적절한 혜택이 돌아오지 않기 때문에 많이 배포되기 어려운 단점을 지니고 있다.

4단계 BASE 메커니즘
지금부터는 기존에 연구된 IP 스푹핑 탐지 기법과는 달리 위에서 설명한 실용적인 측면에서의 세 가지 특성을 만족하면서 네트워크의 중간 경로에서 패킷 필터링을 적용하는 방식에 대해 알아보자.

![그림 4. 4단계 BASE 메커니즘](image)

BASE는 BGP Anti-Spoofing Extension의 약자로, 앞에서 설명한 실용적인 프로토콜이 지나야 할 세 가지 프로토콜의 특성을 만족하는 메커니즘이다. 공격자가 많은 양의 스푹핑된 패킷들을 하나의 목적지로 한꺼번에 보낼 경우 목적지의 네트워크가 과부하가 걸릴 수 있기 때문에 BASE는 목적지 네트워크에 스푹핑된 패킷이 도달하기 전에 미리 그 패킷을 차단한다.

즉 스푹핑된 패킷이 목적지의 호스트에서 출발하여 목적지 호스트로 가는 경로 상에서 BASE 메커니즘을 사용하는 라우터를 만나게 되면 그 라우터에서 그 패킷이 지나야 하는 응답 값을 지나고 있는 값과 동일하다고 가정할 수 있게 된다. 이 경우 BASE 메커니즈가 기존의 경로를 지나검색으로 패킷의 해더(그림 2의 16-bit identifier) 값을 비교한다. 패킷을 전송할 때 사용하는 마킹값은 전송에 BASE 필터 사이에 전달할 때는 오늘날의 인터넷의 표준으로 사용되고 있는 인터-AS 라우팅 프로토콜인 BGP를 이용하여 마킹값을 분배한다. 이러한 BASE 메커니즘은 그림 4에서와 같이 4단계로 구성된다.

1. 마킹값 전달 : 이 단계는 마킹값을 계산하려는 BASE 필터들에 전달하는 단계이다. BGP 메시지는 각각의 BASE 필
다음 페이지의 각각의 펀터들은 이전 라우터들에게 받은 패킷에 적혀있는 마킹값(mi-1)과 자신의 비밀키(ki)를 이용하여 새로운 자신의 마킹값(mi)를 생성한다. 즉, i번째 라우터에서의 마킹값은 \( mi = MAC(ki, mi-1) \)의 값을 가지게 된다. 각각의 BASE 펀터는 받은 마킹값과 자신의 마킹값을 자신의 펀터 테이블에 저장한다. 마킹값을 BASE 펀터들에게 전달하는 이 작업은 BGP 경로가 바뀌지 않는 한 번만 발생한다.

2) 펀터 활성화 : BASE 메커니즘은 항상 동작하는 방식이 아니라 목적지 호스트가 스파핑된 패킷을 받았다고 생각하여 BASE 펀터를 활성화시킬 때만 작동한다. BASE 펀터를 활성화시키는 메시지는 모든 BASE 펀터에게 BGP를 이용하여 전달된다. 각각의 BASE 펀터들이 펀터 활성화 메시지 받으면 그 주소로 가는 펀터들은 정해진 마킹값을 가져 앵 모든 BASE 펀터들로 전파할 수 있다.

3) 패킷 마킹과 펀터링 : 펀터링 단계에서는 각각의 BASE 펀터에서 나가는 패킷들에게 마킹값을 패킷의 헤더에 마킹한다. BASE 펀터로 돌아오는 패킷들은 올바른 마킹값을 가지지 않을 경우 그 패킷들을 드롭시킨다.

4) 펀터 비활성화 : 목적지 호스트가 더 이상 스파핑된 패킷에 대한 위험이 없다고 판단할 경우 BASE 펀터를 비활성화시킨다. 이 메시지 또한 모든 BASE 펀터들에게 BGP를 이용하여 전달된다.

BASE를 이용한 IP 스파핑 방식

앞에서도 설명했듯이 BASE는 위의 과정 중 세 번째 단계에서 패킷에 마킹된 마킹값이 첫 번째 단계에서 BGP를 이용해 분배된 마킹값과 같은 값을 가지면 정상적인 패킷으로 간주되어 다음 라우터로 전달되지만 다른 값을 지날 경우 스파핑된 패킷으로 간주되어 드롭된다. 그림 5에서 볼 수 있는 것처럼 각각의 BASE 펀터는 하나의 펀터링 테워블과 하나의 라우팅 테워블을 가지고 있다. 펀터링 테워블과 라우팅 테워블의 엔트리는 BGP 메커니즘이 모든 라우터들에게 전달되어 질 때 업데이트되면서 완성된다. 이는 BASE의 첫 번째 단계에서 발생된다. 이 상황에서 목적지 라우터가 스파핑된 패킷이 있음을 감지하고 BASE 펀터를 인에 이를 시키고 나면 그 이후에 소스 s가 패킷을 보내고자 할 때 패킷의 헤더 부분에 마킹값을 넣어서 보내야 한다.

이 때 정상적인 소스로부터 생성되어 전달된 패킷의 마킹값은 첫 번째 단계에서 각각의 라우터에 전달되어 펀터링 테워블에 저장되어 있는 마킹값과 같은 값을 지날 것이다. 그림 5에서 라우터 v4가 마킹값을 m2 혹은 m3 중에 어느 하나의 값을 가질 경우 정상적인 경로는 지나온 패킷으로 인식된다.

그 후 그 마킹값이 자신의 비밀키를 연산하여 자신의 마킹값 m4를 만들고 받은 m2나 m3 경로 대신 m4 마킹값을 패킷에 마킹하여 다음 라우터인 v5에 전달하게 된다. 즉 각각의 BASE 펀터에서는 받은 패킷의 마킹값이 펀터링 테워블의 소스 IP 주소에 대응되는 prev_mark에 적힌 마킹값과 같은지를 확인한 후 갑으로 라우팅 테워블의 목적지 IP 주소에 대응되는 next_node로 패킷을 보낸다.

이 때 마킹값은 펀터링 테워블의 소스 IP 주소에 대응되는 next_mark값을 계산한 후 마킹하여 전달하게 된다. 만약 펀터링 테워블에 기록되어 있는 마킹값이 패킷에 적힌 마킹값과 일치하지 않을 경우 이 패킷은 해당 소스 IP 주소를 가진 호스트로부터 거쳐야 할 정상적인 경로를 가지지 않은 패킷으로 간주되어 드롭된다.

BASE는 모든 라우터들이 BASE 메커니즘을 사용하지는 않더라도 설정없이 잘 작동한다. 그림 6은 BASE 메커니즘이 부분적으로 적용되었을 때의 모습을 보여주고 있다. 실패되어 있는 라우터가 BASE 메커니즘을 사용하는 라우터이고 실패되지 않은 라우터는 BASE 메커니즘을 사용하고 있지 않다. BASE 펀터는 중간에 BASE 메커니즘을 사용하지 않는 라우터들이 섞여 있어도 관계없이 그 다음 BASE 펀터와 메시지를 주고받는다.

이 경우에도 그림에서 볼 수 있듯이 라우터 v에서는 m1의 마킹값을 지녀야 정상적인 패킷으로 간주되지만 공격자가 m2로 마킹한 패킷을 보내 보일 경우 BASE 펀터 v에서 걸린다. 또한 그림 6에서처럼 공격자가 BASE 메커니즘을 사용하지 않는 라우터로 패킷을 보내 보일 경우 my로 마킹된 패킷은 목적지까지의 라우팅 경로에서 처음으로 만나는 BASE 펀터에서 스파핑된 패킷으로 간주되므로 드롭된다.

따라서, BASE는 공격자가 스파핑된 소스 IP 주소를 이용해 패킷을 보낼 경우 원래의 소스 IP 주소를 지키는 정상적인 라우팅 경로와 다른 경로를 지나가기 때문에 원래 가져야 하는
장성적인 마진값과 다른 값을 가진 패킷이 BASE 패킷에 도달하게 된다.
또한 각각의 라우터가 마진값을 생성할 때 자신의 비밀키를 사용하기 때문에 공격자는 해당 소스 IP 주소가 가지는 장성적인 마진값을 알 수 없다. 그러므로 스파핑된 패킷을 보면 경우 각각의 라우터에 저장된 패킷 테이블의 값과 다른 마진 값을 가진 패킷이 도달되고 이로 인해 충격적 네트워크에 도달하기도 전에 BASE 패킷에서 스파핑된 패킷으로 간주되어 드름된다.

IP 스파핑 탐지 실험
BASE 메커니즘의 IP 스파핑 탐지 성능 효율성을 측정하고 다른 IP 스파핑 탐지 메커니즘들과 비교하기 위해 Oregon Route View project(http://archive.routeview.org)의 AS connectivity 그래프를 이용하여 실험을 수행했다. 실험에서 사용한 AS 그래프는 2006년 4월의 connectivity이고 약 22,000개의 노드로 구성되어 있다.

그림 6. BASE 패킷의 부분적인 배포 상황에서의 동작

그림 7. 스파핑된 패킷 탐지 실험
스프링된 패킷의 헬링 성능을 비교하기 위해 BASE 뿐만 아니라 ingress filtering, RPF 메커니즘과 함께 실험했다. 그림 7은 스파핑 탐지 메커니즘들의 부분적인 배포 상태에서 각 메체에 따라 헬링 성능이 다르다는 것을 보여주고 있다. BASE가 다른 메커니즘보다 스파핑된 패킷을 탐지하는데 훨씬 좋은 성능을 지니는 것을 확인할 수 있다. 또한 BASE 메커니즘이 앞서 설명한 선행적인 프로토콜이 지나야 할 시 가지 특성을 만족하는 것을 확인할 수 있다.

첫 번째 특성은 초기의 사용자에게로 헬링 효과가 주어져 야 한다는 사항이었는데 그림 7에서 보듯이 전체 AS 라우터의 10%만이 BASE 메커니즘을 사용하더라도 헬링 효과는 매우 큰 것을 볼 수 있다.

두 번째 특성은 정차적으로 향상되는 헬링 효과로서 접점 많은 라우터들이 BASE 메커니즘을 사용함수록 점점 헬링 효과가 증가되는 것을 볼 수 있다. 그리고 세 번째 특성으로 모든 라우터들이 BASE 메커니즘을 사용하지는 않더라도 메커니즘의 반동작이 뒤어난 성능을 보일 수 있다. 따라서 BASE 메커니즘은 3가지의 선행적인 프로토콜로서의 특성을 만족하고 있으며 다른 IP 스파핑 탐지 메커니즘들보다 훨씬 뒤어난 성능을 보임을 알 수 있다.

BASE 메커니즘의 실제 적용
IP 스파핑은 IP 투상자 호스트 이름을 기반으로 한 인증절차를 거치는 보안모델을 공격하는데 효과적으로 사용된다. 이 방법에는 여러 공격 형태의 공격자의 위치를 숨기는 방법으로 사용되는 IP 스파핑 공격의 종류와 위협, 그리고 탐지에 대해 알아보았다.

IP 스파핑은 TCP/IP의 설계와 구현상의 문제점과 기인한 것으로 새로운 프로토콜을 사용하지 않는 한 완벽한 보호 대책은 존재할 수 없다. 하지만 BASE 메커니즘을 이용한 라우터에서의 대응책을 이용하여 스파핑된 패킷이 헬프 호스트에 도달하기 전에 미리 걸려있는 작업을 통해 네트워크의 성능을 상상시키길 수 있다. BASE 메커니즘은 선행적인 프로토콜이 갖추어야 할 세 가지 특성을 모두 만족하는 메커니즘으로 실험자의 배포되어 활용되기 위한 조건을 만족할 뿐만 아니라 다른 메커니즘들과 비교했을 때 스파핑 패킷 탐지에도 효율적임을 보일 수 있었다.