리눅스 기반 IP Spoofing 방지 라우터 프레임웍 설계

박제호, 조은경, 고광선, 엄영익
성균관대학교 정보통신공학부
e-mail:{jhpark, iuno, rilla91}@dclab.skku.ac.kr,
yieom@ece.skku.ac.kr

Design of the Linux Routing Framework for Protecting IP Spoofing

Je-ho Park, Eunkyung Cho, Kwang-sun Ko, Young Ik Eom
School of Information Communication Engineering,
Sungkyunkwan University

요 약

최근 사회적으로 큰 문제로 대두되는 네트워크기반 공격들은 공격 대상 시스템으로 하여금 정상 서비스가 불가능하도록 하는 서비스거부 (분산서비스거부 포함) 공격이 주를 이루며, 이러한 공격에는 기본적으로 패킷의 출발지 IP 주소를 위장하는 IP spoofing 기술이 사용된다. 본 논문에서는 현재까지 상용라우터에 적용되어 있는 IP spoofing 방지 기술과 리눅스시스템에서 제공하는 IP spoofing 방지 기술 그리고 연구논문으로 제시된 IP spoofing 방지 기술을 분석하고, 이를 토대로 IP spoofing 기술을 사용한 공격자의 패킷을 초기에 차단하는 IP spoofing 방지 라우터 프레임웍을 제안한다.

1. 서론

최근 사회적으로 큰 문제로 대두되고 있는 네트워크 기반 공격들은 공격 대상 시스템으로 하여금 정상 서비스가 불가능하도록 하는 서비스거부 (분산서비스거부 포함) 공격이 주를 이루며 이러한 공격에는 기본적으로 IP spoofing 기술이 사용된다. IP spoofing이란 인터넷 네트워크 환경에서 패킷의 출발지 IP 주소를 의도적으로 무의미한 IP 주소나 다른 호스트의 IP 주소로 변경하는 것을 의미하며, 공격자가 자신의 위치가 역 추적되는 것을 방지하기 위하여 혹은 IP 주소로 인증 및 필터링하는 애플리케이션과 시스템을 공격하기 위하여 사용한다. 본 논문에서는 IP spoofing을 방지하기 위한 기존 상용라우터에 적용된 기술과 리눅스라우터에서 제공하는 IP spoofing방지 기술 그리고 최신의 기술 논문상에 제시된 IP spoofing방지 관련 기술을 토대로 IP spoofing 방지 라우터 프레임웍을 제안한다.

본 논문 구성은 다음과 같이 구성된다. 2장에서는 IP spoofing 방지 기술에 대한 관련 연구를 소개한다. 3장에서는 본 논문에서 제안하는 리눅스 기반 IP spoofing 방지 라우터 프레임웍에 대해서 자세히 설명한다. 마지막 4장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련연구

본 절에서는 상용라우터의 IP spoofing을 방지하기 위한 기술, 리눅스라우터에 구현된 IP spoofing 방지 기술과 현재까지 기술논문으로 제시된 IP spoofing방지 기술에 대해 소개한다.

2.1 상용라우터에 적용된 IP spoofing 방지 기술

상용라우터에서는 패킷필터링 기술, MAC 주소와 IP 주소의 바인딩 기술 그리고, unicast RPF(Reverse Path Forward)기술이 있다. 첫째, 패킷 필터링 기술은 ingress, egress 필터링기술을 이용하여 내외부 망간의 비정상적인 패킷을 필터링하는 방법으로 IP spoofing된 패킷을 차단하는 기술이다[1]. 둘째, MAC 주소와 IP 주소의 바인딩 기술은 ISP에서 해당

ISP 가입자에게 할당된 MAC 주소와 IP 주소의 바인딩 내역을 관리함으로써 ISP 사용자 중 기 설정된 바인딩 내역과 다른 MAC 주소와 IP 주소를 사용할 경우 해당 패킷을 차단하는 기술이다[2]. 셋째, Unicast RPF는 라우터가 unicast 포워딩 중 수신한 패킷이 라우터가 관리하는 FIB(Forwarding Information Base)상에 기록된 역 경로정보 중 예상되는 입력 네트워크카드로부터 수신되지 않은 경우 IP spoofing된 패킷으로 간주하여 패킷을 차단하는 기술이다[3].

2.2 리눅스에 적용된 IP spoofing 방지 기술

리눅스에서는 IP spoofing을 방지하기 위해 SAV(Source Address Validation)기능이 구현되어 있다[4]. SAV기능이란 패킷을 포워딩하기 전 단계에 라우트캐쉬에 수신한 패킷에 대한 포워딩 정보가 없는 경우, 해당 패킷에 대한 FIB검색 중 패킷의 유효성을 검사하여 유효하지 않은 패킷을 차단하는 기능이다. 검사하는 항목 중 수신한 패킷에 대한 출발지 주소와의 최단 경로에 존재하는 네트워크카드 정보와 FIB를 통해 역 경로 탐색을 통하여 출발지 IP 주소와 최단 경로에 존재하는 네트워크카드와 다를 경우 spoofing 된 패킷으로 간주하여 차단한다. IP spoofing 방지 관련 리눅스에서의 라우팅 루틴의 세부 내부 함수 호출은 그림 1과 같다.

```
fib_validate_source(infomation of a recived
packet)
{
   key.dst=src;
   key.src=dst;
   get device info and rp_filter info
   if(!lookup(key))   goto error
   if(result device !=
           the device of the received packet)
      goto error
   else
      return true
   ...
}
```

(그림 1) 리눅스의 라우팅 구현부의 IP spoofing처리 함수

그림 1과 같이 fib_validate_source()함수에서는 hash테이블 검색 시 실제 출발지를 목적지로 하는 키를 구성하여 역 경로에 대한 정보를 구한 후 패킷의 유효성을 검사하게 된다.

2.3 기술논문에 제안된 IP spoofing 방지 기술

본 절에서는 현재 IP spoofing을 방지하기 위해 제안된 TTL based hop-count computation 기술, TCP specific 기술, Automatic Spoof Detector 기술을 설명한다[5,6,7].

2.3.1 TTL based hop-count computation 기술

TTL based hop-count computation 기술은 네트워크 계층에 구현할 수 있는 기술로써, IP 헤더에 포함된 'TTL'필드를 사용한다. 기본 원리에 대한 코드는 그림 2와 같다.

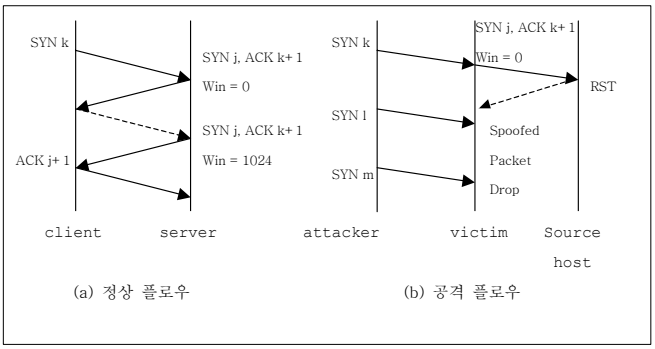
```
for each packet:
   extract the final TTL T and IP address S;
     infer the initial TTL To;
        compute the hop-count Hc = To - T;
   index S to get the stored hop-count Hs;
   if(Hc != Hs)
         packet is spoofed
   else
         packet is legitimate
```

(그림 2) TTL based hop-count computation 기술

그림 2의 알고리즘은 두 특정 호스트 사이의 패킷이 거치는 홉(hop) 수는 일정하며, 만약 TTL값으로 구한 홉 수가 기존의 홉 수와 다를 경우, IP spoofing된 것으로 간주한다는 내용이다.

2.3.2 TCP specific 기술

TCP specific 기술은 TCP 전송계층에 구현할 수 있는 기술로서, TCP 헤더에 포함된 'window size' 필드를 사용한다. 그림 3은 TCP specific 기술의 내용을 설명한다.



(그림 3) TCP specific 기술

그림 3과 같이 spoofing된 패킷을 전송하는 공격자는 공격 대상자의 응답 패킷(ACK)을 확인할 수

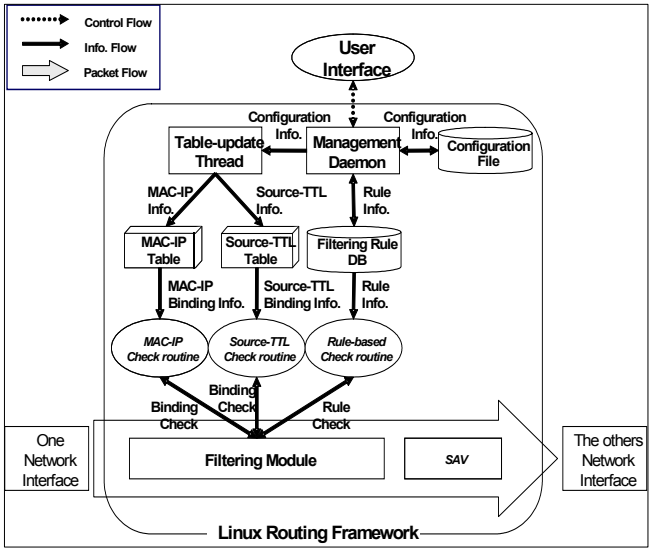
없으므로, 공격 대상자가 흐름 제어를 위하여 응답 패킷의 window size 필드를 변경해도 알지 못한다.

2.3.3 Automatic IP spoofing detector

Automatic IP spoof detector는 네트워크의 특정 호스트에 IP spoofing을 방지하기 위해 설치되는 유틸리티의 일종으로 ARP 요청 패킷을 이용하여 같은 네트워크 상에 존재하는 모든 호스트들의 MAC 주소와 IP 주소의 바인딩 목록을 관리하여 비교함으로써 IP spoofing된 패킷을 차단한다.

3. IP spoofing 방지 라우터 프레임웍

본 논문에서 IP spoofing 방지를 위해 제안하는 리눅스기반 라우터 프레임웍은 4가지 기술을 모듈로 구현한다. 리눅스 시스템에서 IP spoofing을 방지하기 위하여 리눅스 커널에서 옵션으로 제공하는 패킷 필터링과 SAV 기능을 이용하고, 상용 라우터에 적용되어 있는 MAC 주소와 IP 주소를 이용한 바인딩 기술을 이용한다. 그리고 기술 논문에서 제시된 TTL based hop-count computation 기술을 이용한다. 시스템의 전체구성은 그림 4와 같다.



(그림 4) 리눅스 기반 IP spoofing 방지 시스템

프레임웍에서 전반적인 설정 정보와 IP spoofing 방지를 위하여 사용되는 테이블 정보는 그림 4의 management daemon이 관리한다. management daemon은 관리자가 설정한 내용을 적용할 경우 커널 모듈을 커널로부터 분리 한 후 변경된 설정이 반영되도록 다시 커널 모듈을 동작시켜 설정내용을 적용시킨다[8,9]. 커널 모듈로는 그림 4에서와 같이 MAC-IP check모듈, src.-hopcount check 모듈, Rule-based check 모듈이 있으며, 리눅스 시스템에서 제공하는 SAV check모듈로 구성된다. 전체 커널 모듈의 알고리즘은 그림 5와 같다.

```
if (P has no checksum error) {
     if (MAC_IP() is activated) {
          if (MAC_IP(P) is not true){
               P is dropped;
          }
     }
     if (Src_Hopcount() is activated) {
          if(Src_Hopcount(P) is not true){
               P is dropped;
          }
     }
     if (Rule_based() is activated) {
         if (Rule_based(P) is not true){
             P is dropped;
         }
     }
}
if (rp_filter is activated) {
     P is checked by SAV, and P is
     forwarded or filtered.
}
```

(그림 5) 전체 커널 모듈의 알고리즘

그림 5에서와 같이 메인 루틴은 각 세부 커널 모듈을 제어하며, 실질적인 각각의 커널 모듈들의 알고리즘은 그림 6과 같다.

```
/* MAC-IP binding check routine */
int MAC_IP(struct sk_buff P){
     if ((the binding info. of P)
                    ∈  tMAC_IP) {
          return true;
     }
     return false;
}
/* Src.-Hopcount binding check routine */
int Src_Hopcount(struct sk_buff P){
     if ((the binding info. of P)
                    ∈  tSRC_HOPCOUNT) {
          return true;
     }
     return false;
}
/* Rule-based check routine */
int Rule_based(struct sk_buff P){
     Return true or false as the treatment
     policy of P in the dbFILTERING_RULE
}
```

(그림 6) 세부 커널 모듈들의 알고리즘

3.1 MAC-IP check 모듈

MAC-IP check 모듈은 라우터를 거치는 패킷들의 IP 주소와 MAC 주소간의 바인딩 내역을 관리하는 테이블의 정보를 기반으로 모든 패킷을 모니터링한 후 테이블 내역에 위반되는 패킷을 차단한다.

3.2 Src.-Hopcount check 모듈

Src.-hopcount check 모듈은 2.3.1에서 소개된 TTL based hop-count computation 방법을 기반으로 한다. 정상적인 두 호스트 사이의 네트워크 경로에 큰 변화가 없다면 전달되는 패킷이 거치는 홉수는 일정한 값을 가지게 된다는 원리에 따라 라우터는 패킷들의 출발지 IP 주소와 홉 수를 관리하는 테이블의 정보를 기반으로 모든 패킷을 모니터링한 후 테이블 내역에 위반되는 패킷을 차단한다.

3.3 Rule-based check 모듈

Rule-based check 모듈은 ingress/egress에 대한 패킷 필터링 정책을 이용한 필터링 모듈로서, RFC1918에 명시된 지정 IP(사설 IP)에서 들어오는 패킷 차단 기능을 수행하며, RFC2267에 명시된 ingress/egress 트래픽에 대해 위장된 주소의 흐름을 차단하도록 한다[10,11].

3.4 SAV check 루틴

SAV check 루틴은 리눅스 커널에 구현되어 있는 출발지 주소 유효성 검사기능을 사용하는 루틴이다. FIB에 기록된 입력 인터페이스카드로부터 해당 패킷이 수신되지 않을 경우, 해당 패킷은 IP spoofing된 패킷으로 판단하여 차단되게 된다.

4. 결론 및 향후 과제

본 논문에서는 IP spoofing된 패킷을 라우팅하는 동안 차단하는 리눅스기반 라우터 프레임웍을 제안하였다. IP 버전 6에서는 IPsec등의 암호화 통신을 이용하여, 원천적으로 IP spoofing을 방지 할 수 있는 대안을 제안하고 있다. 하지만, IP 버전 6의 상용화 및 보급 확대 이전까지 IP 버전 4에서의 IP spoofing을 이용한 사이버 공격은 계속될 것이며, 그에 대한 대안은 지속적으로 연구되어야 한다. 앞으로 향 후 과제로는 본 논문에서 제안된 시스템에 대한 성능테스트를 통해 보다 최적화된 시스템 프레임웍을 구현하는 것과 기존 IP spoofing 방지 기술과 비교평가 하는 것이다.

참고문헌

[1] http://www.faqs.org/rfcs/rfc1812.html.

[2] Juniper Networks, Ethernet MAC Address Filtering, JUNOS Software Documentation.

[3] Cisco Systems, *Unicast Reverse Path Forwarding*, Cisco IOS Documentation.

[4] http://www.pom.gr/ilisepe1/firewall_help.html

[5] C. Jin, H. Wang, and K. G. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic," *ACM Conference on Computer and Communications Security (CCS '03)*, Washington, DC, Oct. 2003.

[6] S. J. Templeton and K. E. Levitt, "Detecting Spoofed Packets," *Proc. of DARPA Information Survivability Conference and Exposition (DISCEX '03)*, Vol. 1, Apr. 2003, pp. 164 -175.

[7] Y. Jin and S. Walla, *"A Preprocessor Plugin for SNORT: IP Spoof Detect"*, Advanced Network Management Lab, Indiana University, Apr, 2002.

[8] M. Beck et al, *Linux Kernel Programming 3rd Edition*, Addison Wesley, 2001.

[9] D. P. Bovet and M. Cesati, *Understanding the Linux Kernel*, 2nd Edition, O'Reilly, 2002.

[10] http://www.faqs.org/rfcs/rfc1812.html.

[11] P. Ferguson, D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, Network Working Group, January 1998.