Marking Algorithm 기반 IP 역추적의 공격 진원지 발견 기법*

김수덕[0] 김기창 김병룡
인하대학교 전자계산공학과
petitpri[0]@super.inha.ac.kr, kchang@inha.ac.kr, doolyn@super.inha.ac.kr

Marking Algorithm based Attack Origin Detection in IP Traceback

Soo Duk Kim[0], Ki Chang Kim, Byung Ryong Kim
Dept. of Computer Science & Engineering, Inha University

요 약

최근 급증하고 있는 인터넷 사용자들을 위한 인터넷 서비스 업체들의 증가와 더불어 악의적인 공격자들의 공격 또한 증가하고 있다. 이러한 공격으로 인한 인터넷 서비스 업체들에게 치명적일 수 있는 신용에 대한 불신임과 서비스의 불안정이라는 피해는 기업의 이미지를 실추시키는 등 막대한 영향을 끼칠 수도 있다. 이러한 악의적인 공격 형태 중 최근 가장 빈번하게 그리고 큰 피해를 주는 공격 형태가 DoS(Denial-of-Service)[1]공격이다. 그러나 DoS공격에 대한 적당한 대응방법이 아직까지 미비한 상태이고, 공격에 대응하여 방어한다고 해도 그 진원지를 찾아내지 못한다면 추후 동일한 공격자(attacker)에 의해 재차 공격을 받을 가능성을 배제할 수 없는 실정이다. 이에 본 논문은 DoS공격에 대응하는 하나의 방법으로 공격 경로(attack path)를 찾아내고 더 나아가 공격 진원지(attack origin)의 MAC address를 알아냄으로써 공격의 진원지를 찾아내는 방법을 제안한다.

1. 서 론

최근 급증하고 있는 인터넷 사용자의 증가와 이를 위한 인터넷 서비스 업체들의 증가로 인해 이제 인터넷을 이용한 online상의 상거래 및 기업 활동은 일상적이고도 중요한 마케팅의 한 방법으로 자리잡게 되었다. 하지만 상대적으로 중요해진 online상의 기업 활동을 방해하려는 악의적인 공격자들의 공격 또한 증가하고 있으며, 그 피해는 기업이나 인터넷 서비스 업체에게 있어서는 막대한 영향을 줄 수도 있는 것이다. 인터넷을 통한 이러한 기업 활동을 방해하는 악의적인 공격의 대표적인 형태가 DoS공격이다. DoS공격으로 인한 피해는 단순히 server를 마비시켜서 잠시동안 서비스를 중단시키는 정도일 수도 있지만, 인터넷 서비스 업체나 기업에게 있어서 인터넷 서비스의 가장 중요한 요소일 수 있는 신용을 잃게 할 수도 있는 엄청난 피해를 줄 수도 있다. 그러나 DoS공격에 대응하는 적절한 방법이 없는 상황이기 때문에 피해자(victim)의 입장에서는 마비된 server를 재가동 하거나 IDS등을 이용해서 공격을 차단하는 정도의 대응 이외의 다른 대응 수단이 없다는 것이 현실이다. 이에 본 논문에서는 기존의 marking 방식 IP 역추적에서 attack path를 찾아내는 방법을 이해하고 더 나아가 attack origin의 MAC address를 알아내어 그 attack origin의 실제 위치를 찾아내는 방법을 제안한다.

본 논문의 구성은 2장에서 역추적에 관련된 연구 방법들을 알아보고, 3장에서는 기존의 marking 방식 IP 역추적 방법인 compressed edge fragment sampling에 대해서 알아보고 이 방식의 문제점을 제시하며, 4장에서는 기존 marking 방식 IP 역추적의 문제점 중 하나인 attack origin detection 문제를 해결하기 위해 Mac address를 이용한 해결 방법과 그 algorithm을 제안한다. 그리고 6장에서는 제한조건과 문제점을 지적하고, 7장에서 결론과 향후 연구 방향에 대해서 논의할 것이다.

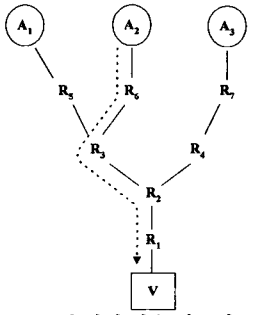
*본 논문은 한국과학기술평가원(KISTEP)의 국가지정연구실사업의 연구비 지원으로 수행되었음. (과제번호: 2000-N-NL-01-C-246)

2. 관련 연구

DoS공격에 대응할 수 있는 적당한 방법을 찾기 어려운 이유는 대부분의 DoS공격이 IP spoofing형태로 진행되기 때문이다. 따라서 packet에 적혀있는 IP로는 공격 진원지를 찾아내는 것이 불가능하다. 이런 문제점을 해결하기 위한 노력으로 Ingress filtering[2], Link testing[3], Logging[4], ICMP Traceback[5] 등의 방법이 연구되고 있다. 하지만 이런 연구들은 각기 단점을 가지고 있으며 그 단점을 극복할 새로운 방법이 연구되어야만 했다.

이에 새롭게 제시된 방법이 marking algorithm[6]이다. 이는 실제적인 attack path를 찾아내어 그 공격을 차단하는 방법으로서, attacker로부터 victim까지의 중간 경로상에 있는 모든 router들이 자신의 IP address를 packet에 marking함으로써 그 packet이 자신을 지나갔음을 표시하는 것이다. 이를 통해 실제적인 attack path를 찾아 낼 수 있다.

[그림 1]은 packet의 여러 전송 경로 중에서 기존 marking algorithm을 통해 찾아낼 수 있는 실제적인 attack path를 나타내고 있다.

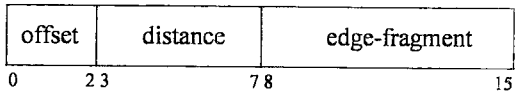


[그림 1] packet의 여러 전송 경로와 attack path

3. Compressed Edge Fragment Sampling Algorithm

Compressed edge fragment sampling algorithm[6]은 router의 정보를 packet의 IP header identification field에 실어 보내는 방식으로 router의 정보는 fragment하여 조각으로 전송하며, 또한 router에서의 marking procedure는 sampling방식에 의해 확률로서 처리된다.

Router가 packet에 자신의 IP address를 marking하기 위해서 IP header의 identification field(16-bit)를 이용한다. [그림 2]는 새롭게 정의된 identification field이다.



[그림 2] 재정의된 IP header의 identification field

각 router는 자신의 IP address(R)와 이 IP address에 not을 사용하여 hash한 hash(R)값을 BitInterleave하여 64 bit의 R'을 생성하며, 이 값을 8개의 조각으로 나누어 그 중 o(offset, random number)번째 조각을 packet에 실어 보낸다. 즉 packet에는 distance와 IP address의 fragment그리고 그 fragment의 위치를 나타내는 offset이 marking되는 것이다. 이렇게 거쳐온 router의 IP address를 표시한 packet들은 victim에서 distance별로 구분되어 조합되고, 이렇게 조합되어 구해지는 IP address가 올바른 router의 IP address로 판단되면 이를 path tree에 기록한다. 이러한 일련의 과정을 통해 path tree를 구성하면 이것이 바로 attack path가 되는 것이다.

하지만 실제적인 attack path를 찾아내서 이를 이용하여 DoS공격에의 대응 방법을 강구 할 수는 있지만, attack origin을 찾아 낼 수 없다는 문제점을 가지고 있다. packet이 지나온 첫번째 router까지의 추적은 가능하지만 여기에서 attack origin을 추적할 수 있는 방법은 없는 것이다.

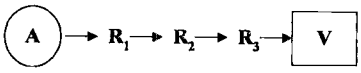
또한 packet이 victim에 도착하기까지 중간에서 각 router에 의해 처음의 정보를 유실(새로운 router의 정보를 marking)하게되는 algorithm의 특징 때문에 첫번째 router의 정보를 가진 packet이 victim까지 도착할 확률이 상대적으로 적어진다. 이 확률은 attacker에서 victim까지의 hop수가 많아질수록 더 작아진다.

4. MAC address를 이용한 attack origin의 detection

4.1 제안하는 algorithm

4.1.1 MAC address의 사용

기존 compressed edge fragment sampling algorithm은 router가 자신의 IP address를 marking함으로써 packet에 자신의 위치를 표시하는 방법을 사용하였다.



[그림 3] 실제적인 attack path

본 논문에서 제안하는 방법은 router가 자신의 IP address뿐만 아니라 바로 앞 단 (previous router)의 MAC address까지도 packet에 marking하도록 하는 것이다. 예를 들면 [그림 3]에서 R1은 자신의 IP address뿐만 아니라 앞 단인 A(attack origin)의 MAC address까지도 packet에 marking하게 되는 것이다.

결국 victim에서 path tree를 구성해보면 attack path의 마지막은 A의 MAC address가 될 것이고, 이를 통하여 attack origin의 정확한 위치까지도 추적이 가능해진다.

4.1.2 확률 p값의 조정

평균적으로 p=0.5로 설정하게 되면 router를 하나 거칠 때마다 처음 정보를 가진 packet의 수가 반으로 줄어들게 된다.

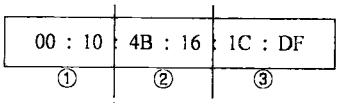
Packet이 router를 지날 때 자신이 가진 정보를 유실하는 것은 어쩔 수 없지만 최대한 많은 packet이 victim에 도착하도록 algorithm을 수정할 필요가 있다.

제안하는 방법은 packet의 distance에 따라서 확률p를 조정하는 것이다. 기존 algorithm에서는 모든 router에서 모든 packet에 대해서 일정한 확률을 설정하지만, 다음처럼 p를 설정하면 기존 정보를 가진 packet의 유실율을 최소화 할 수 있다.

$$p = \frac{1}{(d+1)\times 2}$$

4.1.3 MAC address의 올바른 전송여부를 확인하기 위한 기법

48 bit의 MAC address를 64 bit으로 변형하고 이 64 bit으로 MAC address의 정확성까지 확인하기 위해서 본 논문에서는 다음과 같은 기법을 제안한다.



[그림 4] MAC address

우선 48 bit의 MAC address(M)를 [그림 4]와 같이 세 부분으로 나눈 후 각 16 bit을 XOR(⊕)하여 이 값을 ④로 한다. 여기서 3개의 16 bit을 XOR하는 과정(우변)을 Hash2라 한다.

$$④ = ① \oplus ② \oplus ③$$

이제 16 bit인 ④의 각 bit을 MAC address의 4배수번째 자리에 BitInterleave하여 64 bit의 M'을 생성한다. 이렇게 만들어진 M'을 8개의 조각으로 나누어 packet에 실어 보낸다.

Victim에서 MAC address packet의 조합이 올바른가를 판단할 때는 위의 과정을 역으로 수행하여 ④ 와 ① ⊕ ② ⊕ ③의 값이 같은지를 검사해 본다. 만일 두 값이 같다면 이 MAC address는 올바르게 전송되어 제대로 조합된 값임을 인정한다.

4.2 Marking procedure at router R

[그림 5]는 각 router에서 packet에 IP address와 MAC address를marking하는 procedure를 나타내는 algorithm이다.

```
Marking procedure at router R:
    let R' = Bitinterleave(R, Hash1(R))
    let M' = 4th-Bitinterleave(M, Hash2(M))
    let k be the number of non-overlapping fragments in R'
    for each packet w
        let x be a random number from [0..1)
        let y be a random number from [0..1)
        if x < p then
            if y < 0.5 then
                let o be a random integer from [0..7]
                let f be the fragment of R' at offset o
                write f into w.frag
                write 0 into w.distance
                write o into w.offset
            else
                let o be a random integer from [0..7]
                let f be the fragment of M' at offset o
                write f into w.frag
                write 1 into w.distance
                write o into w.offset
        else
            if w.distance = 0 then
                let f be the fragment of R' at offset w.offset
                write f⊕w.frag into w.frag
            2 increment w.distance
```

[그림 5] 제안하는 marking procedure algorithm

여기에서 x<p인 경우 router는 다시 random number y(0≤ y<1)를 발생시켜서 이 값이 0.5보다 작은 경우는 자신의 IP address를 marking하는 procedure를 수행하게 하고, 이 값이 0.5보다 큰 경우에는 앞 단의 MAC address를 marking하는 procedure를 수행하게 한다.

다음으로 [그림 6]은 victim에서 동일한 distance를 가지는 packet들을 조합하여 검사하고, 올바른 packet일 경우에는 IP address와 MAC address를 가지고 path tree G를 구성하는 algorithm이다.

```
Path reconstruction procedure at victim v:
  let FragTbl be a table of tuples (frag, offset, distance)
  let G be a tree with root v
  let edges in G be tuples (start, end, distance)
  let maxd := 0
  let last := v
  for each packet w from attacker
    FragTbl.Insert(w.frag, w.offset, w.distance)
      if w.distance > maxd then
        maxd := w.distance
  for d := 0 to maxd
    for all ordered combinations of fragments at distance d
      construct edge z
      if d≠0 then
        z := z ⊕ last
      if d := even-number then
        if Hash(EvenBits(z)) = OddBits(z) then
          insert edge (z, EvenBits(z), d) into G
          last := z
      else
        if Hahs2(z) = 4th-Bits(z) then
          insert edge (z, M(z), d) into G
  remove any edge (x, y, d) with d≠distance from x to v in G
  extract path (R_i .. R_j) by enumerating a cyclic paths in G
```

[그림 6] 제안하는 reconstruction procedure algorithm

5. 결 과

5.1 packet 전달율의 증가와 그 편차의 감소

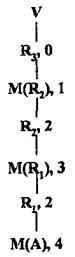
[표 1] packet 전달율 (단위 %)

		R_2	R_3	R_4	R_5	R_6
packet 전달율	d=0	50	50	50	50	50
	d=1	50	25	25	25	25
	d=2		25	12.5	12.5	12.5
	d=3			12.5	6.25	6.25
	d=4				6.25	3.125
	d=5					3.125

[표 1]은 각 router에서 marking한 packet의 전달율을 나타낸다. 여기서 볼 수 있듯이 기존 algorithm에서는 distance가 커질수록 packet의 전달율이 급감했지만, 제안하는 algorithm에서는 큰 distance를 가지는 packet의 전달율이 현저히 높아지고, 각 packet의 전달율 편차가 줄어든 것을 볼 수 있다.

5.2 Attack Origin Detection

[그림 3]의 각 router들이 제안한 algorithm을 수행할 수 있다면 victim V에서는 각 router에서 marking된 서로 다른 distance를 가진 packet들을 받아들이게 되고, 이를 [그림 6]의 과정을 통하여 path tree를 구성해 보면 [그림 7]과 같은 결과를 얻어 낼 수 있다.



[그림 7] 완성된 path tree

[그림 7]에서 마지막 edge인 M(A)는 attack origin인 A의 MAC address를 나타내고, MAC address를 나타내는 edge의 distance는 victim으로부터의 거리를 나타내는 것이다. 결국 attacker가 distance 4의 위치에서 router R_1, R_2, R_3를 거쳐 victim을 공격했다는 것을 알 수 있게 된다.

6. Limitation

- MAC address spoofing에 대하여 무력화.
- 추적 가능한 router hop수의 감소.
- 발견된 origin에 실제 attacker가 존재할 가능성을 확신할 수 있는 확률과 그 이후 추적 방법의 미비.

7. 결론 및 향후 연구 방향

제안하는 algorithm을 통하여 attack origin을 찾아 낼 수 있게 됨으로써 기존에 attack path를 찾아내어 DoS공격에 대응하던 소극적인 대응에서 한발 더 나아가 attack origin에까지 그 대응 범위를 넓힐 수 있게 될 것이고, 이를 통해 추후 재공격의 가능성도 제거할 수 있게 된다. 또한 발견되는 attack origin이 1차적인 attacker의 점유 기점이라고 할 때, 실제적인 attacker의 위치를 찾아 나갈 수 있는 기초를 제공한다는 점에서 본 연구의 의의를 둘 수 있다.

향후 연구는, A에서 다시 attacker의 실제 위치를 찾아낼 수 있는 방법을 연구해야 할 것이고, attacker의 어떠한 packet 조작에도 무력화 되지 않을 강력한 역추적 방법을 찾아 낼 수 있어야 할 것이며, 또한 추적 가능한 hop의 수를 늘리는 방법도 연구가 되어야 할 것이다.

참고 문헌

[1] Computer Emergency Response Team (CERT), "CERT Advisory CA-2000-01 Denial-of-service developments," Jan. 2000, http://www.cert.org/advisories/CA-2000-01.html

[2] P. Ferguson and D. Senie. "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", RFC 2267, Jan. 1998.

[3] R. Stone. CenterTrack: An IP Overlay Network for Tracking DoS Floods. In to appear in Proceedings of thje 2000 USENIX Security Symposium, Denver, CO, July 2000.

[4] G. Sager. Security Fun with Ocxmon and Cflowd. Presentation at the Internet 2 Working Group, Nov. 1998.

[5] S. M. Dellovin. ICMP Traceback Messages. Internet Draft: draft-bellovin-itrace-00.txt, Mar. 2000.

[6] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Practical network support for IP traceback," in Proc. of ACM SIGCOMM, pp. 295-306, Aug. 2000.