現有IP Spoofing 及ARP Spoofing 偵測與防禦技術之實作與分析

SHANG YI-CHIN
台大資工系
Taiwan

LIU HOU-CHEN
台大資工系
Taiwan

CHEN FU-CHUN
台大資工系
Taiwan

LIN CHIEN-HUNG
台大資工所
Taiwan

Abstract

現今已有許多針對IP Spoofing、ARP Spoofing 偵測、防禦的方式與技術，並且在SDN 架構下，又有許多的變化與進展，然而卻缺少這些技術之間的比較與整理，因此我們針對不同的防禦方式建立結構相似的網路拓墣，在相似的環境之下發送攻擊封包並實作偵測防禦，利用成功分辨攻擊封包的比率或是performance 的好壞進行結果分析。

在IP spoofing 這個部分，我們實作了兩種防禦方式，分別是利用hop count value 進行封包的過濾，以及利用SDN 進行封包的過濾；而在ARP spoofing 的部分，我們也實作了兩種防禦方式，分別利用unicast 的傳送ARP request 以及利用SDN，兩種方式皆利用到DHCP server。

Keywords: Network Spoofing, SDN

ACM Reference Format:
SHANG YI-CHIN, LIU HOU-CHEN, CHEN FU-CHUN, and LIN CHIEN-HUNG. 2021. 現有IP Spoofing 及ARP Spoofing 偵測與防禦技術之實作與分析. In *Proceedings of CNS*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Motivation

由於IP spoofing 和ARP spoofing 是許多攻擊方法的前置手段之一，如DDoS 攻擊，中間人攻擊等等，因此能有效防禦IP spoofing 和ARP spoofing 的方法在現今越來越龐大的網路環境中是相當重要且需要的，如果能成功防禦住IP spoofing 以及ARP spoofing 的話，其後續的攻擊便會變得難以實施，增加發起攻擊的困難度。

我們希望透過實作論文中針對IP spoofing 和ARP spoofing 的防禦方式比較不同方式的防禦效果、部署成本、適合的使用情境等等，並根據模擬的結果分析這些方式各自的優缺點及其限制。

CNS, June 2021,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.2 IP Spoofing Defense: Hop Count Filter

這篇論文[4] 提出的方法是利用每個封包在經過路由器時，封包內的TTL 值都會減少一，因此server 可以利用收到封包內的TTL 值算出此封包從client 到server 總共經過幾個路由器，也就是該封包的Hop Count ，而server 可以將不同IP 位址對應的Hop Count 記錄下來，建立並維護一個IP to Hop Count table，以此形成部署於server 的Hop Count Filter (HCF)。當攻擊者只偽造攻擊封包的source IP 的話，該攻擊封包的Hop Count 值就會和合法封包的Hop Count 值有出入，server 就能藉由其HCF 將異常封包過濾掉，達到防禦IP spoofing 的效果。

1.3 IP Spoofing Defense: IP Filter with SDN

從這篇論文中[3]，如果要防止IP Spoofing 最直觀的方式，就是針對流入的封包中的來源IP 進行核對，如果進來封包的來源IP 並不是路由器底下所涵蓋的IP 就會將封包丟棄，反之將封包傳出去。

但是這樣的方法會面臨到一些問題，首先像是在越上層的路由器或交換機就越難知道底下的IP 分布，以及變動的網路會增加管理者的負擔，第二，filter 本身的部屬率會影響到實際IP spoofing 的防禦成功率，但是管理者部屬filter 對於管理網路的保護幫助並不大，因此反而降低了管理者部屬的意願，另外，正常來說在傳統網路上，L2 交換機是無法獲得IP 的資訊，所以在domain 內的IP spoofing 是無法防禦的。結合以上幾點，雖然單純的filter 是簡單有效的方式，但是卻不能有效的完全阻擋IP spoofing 的問題。

基於以上幾點，在SDN 底下由於集中式的controller 擁有網路內的封包資訊，不只是擁有整個網路的概觀資訊，並且可以對網路中的flow 進行處理，進而達到在交換機中建立filter 的功能，同時因為可以獲得不只是MAC 、IP，甚至是應用層的資訊，例如DHCP，來建立對IP spoofing 更好且易於部屬的防禦方式。

因此我們想分析filter 本身的部屬率對IP spoofing 成功率的影響，以及通過[2] 的演算法的方式來降低filter 的部屬但盡量最大化IP spoofing 的保護，並且通過引入SDN 的技術，看能不能更好的達到防禦IP spoofing 的目的。

1.4 ARP Spoofing Defense: S-UARP

從這篇論文[1]當中，我們可以知道，由於DHCP server 負責發送IP，因此他會有正確的IP-to-MAC 的mapping，此論文利用這個特性提出了一個新的方法，S-UARP。全名為Secure Unicast Address Resolution Protocol，當host A 想要與host B 溝通時，原先A 會發送broadcast 的arp request 封包向所有人詢問，然而這樣的方式就會讓attacker有機可趁，而S-UARP 便針對這點做改進。首先，論文提出一種新的DHCP server，DHCP+，他會存所有的IP-to-MAC 的mapping，當A 想與B 溝通時，A 首先會向DHCP+ 發送S-UARP request 詢問B 的MAC address，接著DHCP+ 將會發送response 給A，當A 收到request 時並驗證正確性之後，便向DHCP+ 發送ACK message 並且將正確的MAC address 更新，最後與host B 進行溝通。詳細過程如下：

$A \rightarrow DHCP+ : S - UARP_{req}$
$DHCP+ \rightarrow A : S - UARP_{res} + MIC$
$A \rightarrow DHCP+ : (ACK)K_{sa}$

A: host A
$DHPC+$: DHCP+ server
$S - UARP_{req}$: S-UARP Request 封包
$S - UARP_{res}$: S-UARP Response 封包
MIC: Message Integrity Code（用於驗證封包正確性）
K_{sa}: host A 與DHCP+ server 之間的shared key

1.5 ARP Spoofing Defense: SDN

這個方法也是利用DHCP server 負責發送Local IP，所以DHCP OFFER broadcast 應該會有正確的IP-to-MAC 的mapping，switch 可以過濾所有經過他的DHCP OFFER broadcast，回傳給controller 並把它紀錄在IP-to-MAC 的table 上，當host A 想要與host B 溝通時，因為host A 只知道其IP 不知道其MAC，會先發送arp request broadcast 向所有人詢問，此時attacker 便可假裝自己是host B 回傳錯誤的arp reply 造成原該送往host B 的封包全部送往attacker，attacker 可以再把這些封包作IP spoofing 轉送給host B 實現中間人攻擊，但因為現在controller 上部屬了IP-to-MAC table，switch 可以把所有的arp broadcast 送給controller 作檢查，只有arp broadcast IP maps MAC 時才可送出，否則就修改甚至攔截掉明顯惡意的arp reply，讓attacker 無法成功執行ARP Spoofing。

1.6 Contributions

針對IP Spoofing 以及ARP Spoofing 我們各選擇了兩種防禦方法實作與分析，並在現有的一些方法下進行改進與優化，細節部分將會在Results 的區塊說明實作的步驟以及成果的展示。另外，針對兩種Spoofing 我們皆刻意選擇一種沒有利用到SDN 的方法，與另外一種有利用到SDN 的方法，藉此比較有無SDN 之間的差異。

2 Problem definition

2.1 Problem

現今已有許多針對IP Spoofing、ARP Spoofing 偵測、防禦的方式與技術，並且在SDN 架構下，又有許多的變化與進展，然而卻缺少這些技術之間的比較與評估，因此我們希望能透過自行建立的網路架構，實作一個用來評估的系統，針對兩個攻擊，分別模擬好幾種不同的偵測及防禦方法。

2.2 Attacker Model

作為攻擊者，我們可以任意竄改要傳出的封包內容，或是傳出我們不應該傳出的封包，因此我們能夠很輕易的達到IP 和ARP spoofing 的攻擊，但是要完全防禦IP 和ARP 的spoofing 攻擊卻相對複雜不少，因此我們考慮在同一個Domain 底下，除了Victim 本身以外，所有Hosts 都有可能成為攻擊者，並且攻擊者能達到:

- 攻擊者可以建構任意的封包，並發送到網路上。
- 攻擊者可以偽造成特定或隨機的IP source address，來進行IP spoofing 的攻擊。
- 攻擊者可以使用自己的MAC address 來偽造ARP 封包以竄改或毒害主機的ARP table，達成ARP spoofing 的攻擊。

在這樣的環境下，分析各個防禦方式的成效。

3 Results

3.1 System Design

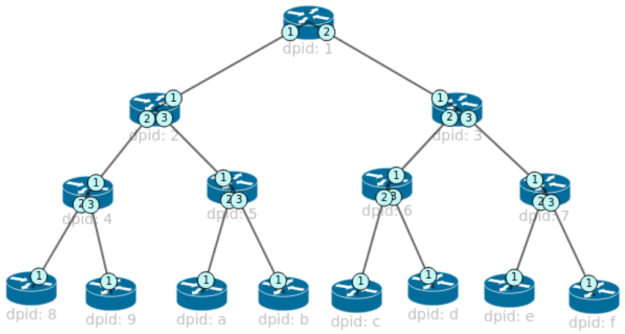

Figure 1. 網路架構圖

在實驗中我們的網路拓樸會使用如上圖的Complete Binary Tree 的架構，其中網路環境是透過Mininet 來進行模擬，在SDN Controller 的部分，我們則是採用Ryu Controller，模擬攻擊者偽造封包是透過scapy 或packit 來達成。

3.2 IP Spoofing Defense: Hop Count Filter

我們在不同hosts 數量的情況下將HCF 部署於server 端，hosts 數量為4, 8, 16, 32 ，每個host 會以相同的機率送出合法或者是攻擊封包，其中攻擊封包只會將source IP 更改為其他host 的IP，不會修改封包的TTL 值。我們最後將HCF 的模擬結果分別繪圖，模擬結果為以下圖2及圖3。

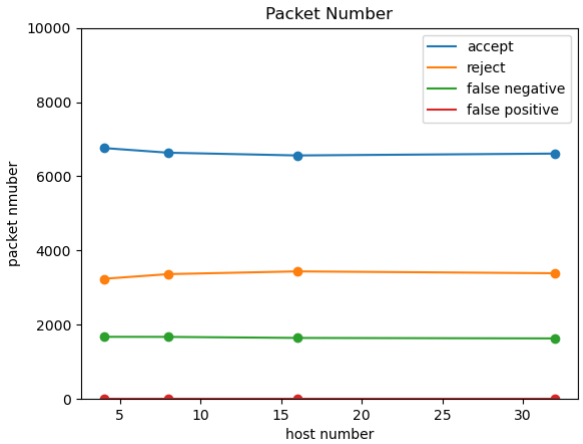

Figure 2. 封包數量

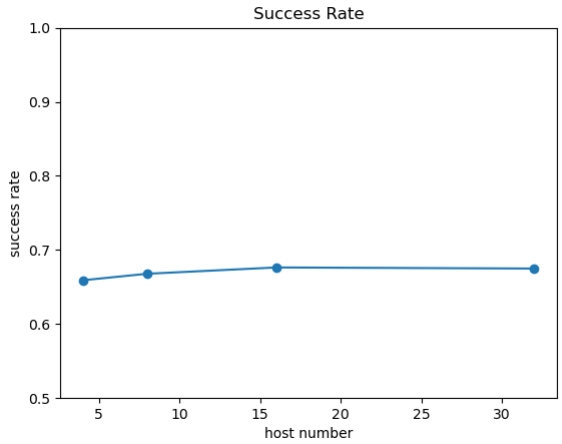

Figure 3. 成功率

圖2為server 收到不同封包的數量，圖3為HCF 的成功率。如兩張圖所示，HCF 即使在hosts 數量不同的情況下，表現結果都差不多，成功率會落在大約66.7%附近，之所以會有這樣的結果是因為我們採用的網路架構是complete binary tree 的緣故。

針對HCF 在complete binary tree的網路架構中的表現，我們可以由以下計算確認其結果：設P'為server 接受一個偽造封包的機率，h為樹的高度，n為host 的數量，則

$$P' = \frac{1}{(n-1)(n-2)} \sum_{i=0}^{h-1} 2^i(2^i - 1)$$
$$= \frac{1}{(n-1)(n-2)} (\sum_{i=0}^{h-1} 4^i - \sum_{i=0}^{h-1} 2^i)$$
$$= \frac{1}{3}$$

成功率則為$1 - P' = \frac{2}{3}$。

我們可以觀察到HCF 的成功率會和hosts 的Hop Count 值有關，如果有很多hosts 有相同的Hop Count 值，那麼HCF 將會無法分辨攻擊封包和正常封包，導致成功率降低；而當不同hosts 有不同的Hop Count 值時，利用HCF 來防禦IP spoofing 則可以得到不錯的防禦效果。

此外，由於Network Address Translation (NAT) 的技術會使得單一IP 位址擁有多個不同但合法的Hop Count 值，如果HCF 沒有考慮到NAT 的存在，server 可能會誤將合法封包視為攻擊封包，導致false positive 的產生，因此當server 在使用HCF 防禦IP spoofing 時必須額外考慮到這一點。

3.3 IP Spoofing Defense: IP Filter with SDN

首先在這裡我們想分析filter 的部屬率，實際上對IP spoofing的防禦成功率造成的影響，我們會在眾多的hosts 中挑選其中一個作為Server，其他hosts 會對Server 進行IP spoofing 的攻擊，攻擊成功的情況也就是成功的將封包送到Server，失敗就是中途就被交換機阻擋了，藉此來分析防禦的成效。

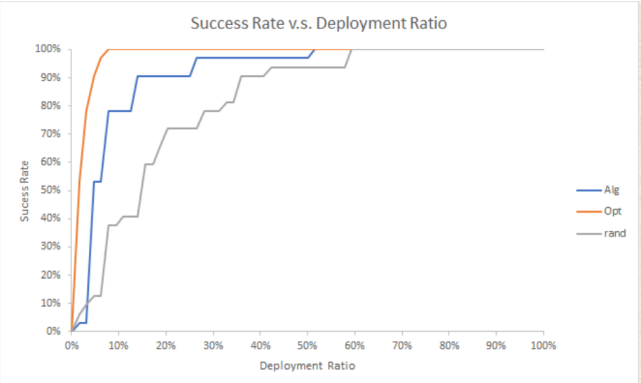

Figure 4. 部屬率與防禦成功率分布圖

圖4是防禦的成功率與filter 部屬率的相關圖，每一輪我們會對一台機器部屬filter，並讓所有hosts 對server 進行IP spoofing 的攻擊，來分析防禦的成功率。灰線是我們每一輪隨機部屬filter 於一台交換機上，大概到達60%才能夠完全抵禦IP spoofing 的攻擊，藍線是我們透過[2]提到在intra domain 內的greedy 演算法，演算法的挑選會從IP 涵蓋範圍大的，也就是越上層的交換機開始部屬，但是考量到我們模擬攻擊的環境，因為我們是固定的一台Server，所以我們就再進一步的優化演算法，得到橘線。

假設現在我們只有一台要保護的Server，因為拓樸是樹狀架構，我們可以不考慮有環的產生，攻擊者要傳送封包到Server 的路徑勢必會包含一個subtree，只要對攻擊者在這個subtree 中的任一ancestor 做部屬，就能防止這個攻擊者的偽造封包，其中subtree 的root 涵蓋的範圍最廣，所以要最小化挑選的交換機，但最大化成功率就是挑選subtree 的root。當有多台Server，就是對多個subtree 進行交集，並挑選最上層的ancestor 的交換機

做部屬，最極端狀態就是要保護所有的hosts，所以在每台End 端交換機做部屬需要$2^{log(n)}$ 台交換機，如果只要保護一台Server 就只要$log(n)$ 台交換機。
前者IP filter 方式是屬於靜態的方式，必須清楚每個交換機底下的IP 範圍，如果交換機底下hosts 的IP 不斷的變動，而非固定的，前者的方式就可能會出問題，因此我們想透過SDN 的技術，透過DHCP protocol 來達到動態的IP filter。
當controller 獲得DHCP Offer 的封包，撈出Yiaddr 來得到host 的IP，並直接在End 的交換機部屬filter，並在期限或是DHCP release 的時候去解除filter，來達到動態配置filter 的功能。。 因為我們是在End 端的交換機進行動態的部屬，所以在所有hosts 都要有IP 情況下，就會對$2^{log(n)}$ 台交換機部屬filter，但是相對的這本身就能對所有hosts 進行IP spoofing 的保護，假設我們要特別針對某些hosts 進行保護的話，也可以採用Method 1 的演算法來減少實際需要部屬的filter，但最好的優點是，我們可以通過DHCP 來達到動態、自動化的filter 來防止IP spoofing。
但在我們的方法要使用dynamic filter 是要信任於我們的DHCP server，並且Controller 要有能力去得到DHCP 封包，我們可以從兩個方向來討論，其實網路管理者是可以將DHCP Server 架設在Controller 內部的，在這樣的情況下，Client 要拿取IP 勢必會通過Controller 進而對交換機部屬filter，作為犧牲可能會對Controller 增加一些負擔和可能會增加一些安全疑慮。第二種情況是Controller 和DHCP Server 分開，一般而言Controller 在自己的domain 底下是不會任意出現DHCP Server 的，因為Controller 本身是可以控制將所有DHCP flow 不准通行的，並且DHCP Server 通常也是網路管理者所架設，假設信任DHCP Server 應該是合理的，如果現在有一個加密的DHCP protocol 的話，Controller 本身可能也必須要有DHCP Server 的密鑰，基於兩者都屬網路管理者管控，我認為是合理的。

3.4 ARP Spoofing Defense: S-UARP

我實作的方法是：首先，利用python script 將正確的IP-to-MAC mapping 存入代表DHCP+ 的host 當中；接著發動攻擊，利用attacker host 發送spoofed arp reply 封包將victim host 的arp cache 寫入錯誤的MAC address；當victim host 想要與其他的host 溝通時，便會先向DHCP+ 發送S-UARP request 封包詢問正確的MAC address，當接收到S-UARP reply 封包且驗證過後，victim host 將會把正確的MAC address 寫入自己的arp cache 當中，再向DHCP+ 發送ACK message。如此一來，即便victim host 最初的arp cached 是poisoned 的狀態，他仍舊能透過S-UARP 拿到正確的MAC address 以進行正確的傳輸。
利用S-UARP 的方式可達到兩種優化的效果：

- 1. Packet reduction：不同於一般ARP request 的封包（broadcast），S-UARP 是host 利用unicast 的方式去向DHCP server 詢問接收者的MAC address，如此一來，在不考慮有ACK message 的情況下，S-UARP 的總封包數量僅僅是一般ARP reply 封包數量的兩倍。下圖為論文當中的分析結果，我們可以清楚地看到S-UARP 的封包數量明顯的小於一般的ARP scheme 或是其他的ARP scheme。

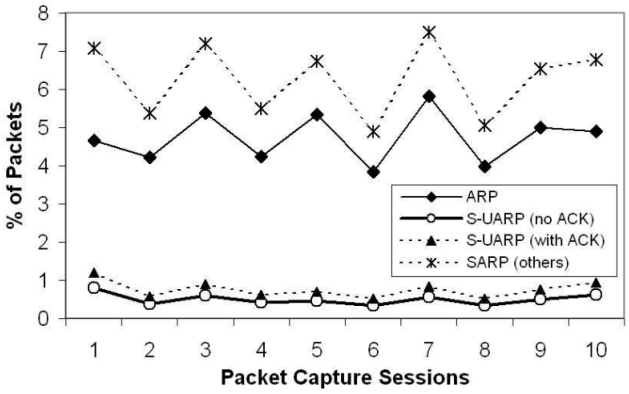

Figure 5. Packet reduction

- Less time consumption：下圖為論文當中的分析結果，ARP 代表一般的ARP scheme，而S-UARP_1、S-UARP_2、S-UARP_3 則是代表不同encryption scheme 的S-UARP（一至三為難度低到高）。由統計圖表可知，S-UARP_1 以及S-UARP_2 所花費的時間都較一般的ARP scheme 少。

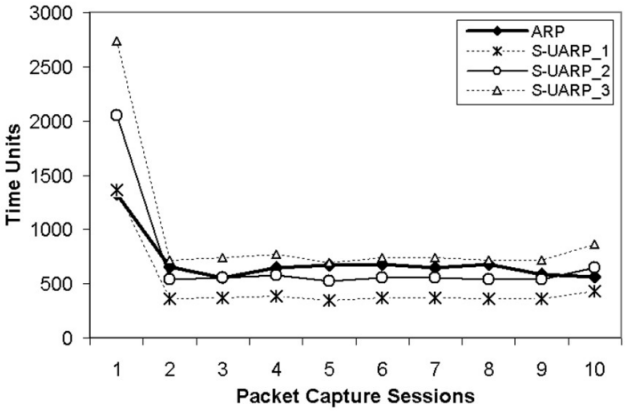

Figure 6. Less time consumption

3.5 ARP Spoofing Defense: SDN

這個方法雖然可以抵擋ARP Spoofing Attack，但有一些顯而易見的缺點，第一就是在論文之中也有研究的DETECTION AND MITIGATION TIME，作者發現隨著Complete Binary Tree 高度的增加，ARP broadcast 必須經過更多的switch 才能到達目的地，若部屬這樣的SDN switch 在整個網路之中，每經過一次switch 都必須回傳給controller 進行檢查，造成同一個封包可能最多被檢查O(log n) 次，每次檢查時間根據網路狀態不同而異，本實驗約莫花費10 ms 的時間，當網路複雜時會造成很

Table 1. 所需Filters 比較圖

Type	A Best Case	Worst Case
Static	$log(n)$	$2^{log(n)}$
dynamic	$2^{log(n)}$	$2^{log(n)}$

大的ARP broadcast latency。第二是DHCP Spoofing，攻擊者可能可以假裝自己是DHCP server 亂發DHCP OFFER broadcast 使的controller 記錄錯誤的資訊，attacker 甚至可以影響正確的ARP broadcast 傳送，解法是像上述的方法讓controller 與DHCP server 建立具有authorization and integrity 的連線，第三是ARP flooding attack，額外的紀錄資料代表attacker 有機會直接塞爆它，不過如果attacker 可以瘋狂更換MAC 跟DHCP server 要求新的IP，我想應該是DHCP server 會先遇到類似(d)dos 的問題。而最後教授提到網路中controller 可能由不同人所控制，真實的網路確實不可能由一個人控制所有switch，不過此方法controller 之間並不會有衝突，部屬防禦的SDN switch 在網路中的覆蓋率與節點選擇才是影響防禦效果的最大因素，然而雖然不同controller 之間不需要溝通或交換資料，方便各自部屬防禦，但說實話對一般使用者來說並沒有太大的誘因。

4 Related work

[2] 分別提出了在inter-domain 以及intra-domain，通過SDN 的防禦方式，其中inter-domain 是通過Source Address Validation 的方式，intra-domain 是通過Source IP filter 也就是我們主要探討的部分，其中intra-domain 作者提出了一個heuristic 的演算法，目標是盡量減少SDN 的交換機，但不損害IP filter 的功用。

[5]提出StackPi 的封包標記方式，及新的過濾機制，藉由封包經過不同路由器會得到不同的路徑標示(Path Identification, Pi)，藉此偵測IP spoofing 的攻擊封包。

5 Conclusion and future work

我們實作並分析針對IP spoofing 和ARP spoofing 兩種攻擊各兩種的防禦方式，傳統網路因為硬體架構下，常常只能通過對protocol 或是想辦法在應用端盡可能獲取資訊來做防禦，往往這樣的解法會相對複雜不少，但在SDN 技術的引入下，其動態控制以及網路概觀資訊下，達成在傳統網路無法輕易達到的功能。

但實際的網路環境中，可能不只有IP spoofing 和ARP spoofing 的攻擊，例如DNS spoofing 或是跟無線網路相關的MAC spoofing，傳統網路中可能會藉由新的硬體或是新的protocol 解決，如果在SDN 的參與下，可能會有更好的解法，可供未來研究。

6 References

[1] B. Issac. (2014). Secure ARP and Secure DHCP Protocols to Mitigate Security Attacks.

[2] C. Zhang et al., "Towards a SDN-based integrated architecture for mitigating IP spoofing attack", IEEE Access, vol. 6, pp. 22764-22777, 2018.

[3] Daniel Senie and Paul Ferguson. (2000). Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827. 10.17487/RFC2827.

[4] Haining Wang, Cheng Jin, and Kang G. Shin, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," IEEE/ACM Trans. Networking, vol. 15, no. 1, Feb. 2007.

[5] A. Yaar et al., "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense", IEEE Journal on Selected Areas in Communications, vol. 24, Issue: 10, Oct. 2006.