HAL
open science

Detecting IP Spoofing by Modelling History of IP Address Entry Points

Michal Kováčik, Michal Kajan, Martin Žádník

To cite this version:

Michal Kováčik, Michal Kajan, Martin Žádník. Detecting IP Spoofing by Modelling History of IP Address Entry Points. 7th International Conference on Autonomous Infrastructure (AIMS), Jun 2013, Barcelona, Spain. pp.73-83, 10.1007/978-3-642-38998-6_9 . hal-01489972

HAL Id: hal-01489972

https://inria.hal.science/hal-01489972v1

Submitted on 14 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Detecting IP spoofing by modelling history of IP address entry points

Michal Kováčik[1], Michal Kajan[1], Martin Žádník[2]

[1] IT4Innovations Centre of Excellence
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, Brno, Czech Republic
ikovacik,ikajan@fit.vutbr.cz
[2] Cesnet
Zikova 2, Prague, Czech Republic
zadnik@cesnet.cz

Abstract. Since a lot of the networks do not apply source IP filtering to its outgoing traffic, an attacker may insert an arbitrary source IP address in an outgoing packet, i.e., IP address spoofing. This paper elaborates on a possibility to detect the spoofing in a large network peering with other networks. A proposed detection scheme is based on an analysis of NetFlow data collected at the entry points in the network. The scheme assumes that the network traffic originating from a certain source network enters the network under surveillance via a relatively stable set of points. The scheme has been tested on data from the real network.

1 Introduction

The source IP address spoofing is inherently used during attempts to hijack network sessions [19] or to scan a target in a stealth mode [9]. But most commonly, spoofing plays an important role during denial-of-service attacks (DoS) or distributed denial-of-service attacks (DDoS). The goal of these attacks is to exhaust network or host resources by flooding the victim with an overwhelming number of packets. As a result, the service provided by the victim becomes unavailable. The spoofing is used to:

- generate large amount of new connections,
- hide the true source identity and render filtering the source of an attack very hard,
- amplify and/or reflect the attack to the victim.

Therefore, there should be counter-measures to prevent IP spoofing or at least procedures to trace back the true source. Although many research has been done in this area none of the proposed solutions became deployed widely. This is of no surprise since the spoofing might be largely mitigated by installing ingress filtering in the stub networks, yet this is rather an exception than a rule.

In our work, we propose an algorithm to detect occurrence of the flows with spoofed IP addresses. We consider network operator (Tier-1, Tier-2) with peering interconnections to other large networks. The scheme works upon NetFlow v5 [16] data collected from the entry points in the operator network.

The scheme is based on the following assumptions:

- there is a set of specific source IP addresses that should not appear in the packets entering the network,
- large portion of the communication is symmetric (i.e. it takes the same path from source to destination and vice versa),
- network traffic originating from a certain network enters the observed network via stable set of points,
- the number of new source IP addresses is stable.

It is straight-forward to build a classifier based on the first assumption. Unfortunately, the assumption covers only a limited set of traffic with potentially spoofed addresses. The second assumption allows to verify legitimate traffic but cannot detect spoofing itself. The third assumption allows to report on which link and from which source prefix the spoofing occurs whereas the last assumption allows to report the destination prefix of the traffic with the spoofed source. The proposed scheme provides several outputs which may serve as an additional information for anomaly or attack detection methods as well as a basis for filtering decisions or post mortem forensics.

The rest of the paper is organized as follows. Section 2 discusses related work on IP spoofing prevention and traceback. Section 3 proposes an algorithm for IP spoofing detection with the use of NetFlow data. Section 4 provides information about the deployment and achieved results. The last section sums up the paper and discusses further research directions.

2 Related work

As previously mentioned, IP spoofing plays an important role in some types of attacks and a lot of research interest has been paid to study methods for preventing or tracing back spoofed IP addresses.

A basic preventive method suggests an ingress filtering [4] in customer or source ISP networks where the pool of legitimate source IP addresses is well known. In order to allow filtering in transit or destination networks, the information about legitimate source IP addresses must be passed from a source towards the destination networks. In [8], authors propose a new protocol to spread this information to routers along the path which may build a filtering table for each ingress interface accordingly. The SPM [2], which is an alternative method designing a scheme in which participants (involved autonomous systems – AS) authenticate their packets by a source AS key. In fact, a host may spoof an IP address from within the same subnet in all these schemes. To address this problem, Shen et al. [13] extended SPM into the intranet where a host tags its packets to authenticate them for a network gateway. Xie et al. [20] proposed

authentication of a host connecting to the Internet by an established authentication protocol. TCP SYN cookies, improved in [21], may be considered as an IP-spoofing prevention although the method works only for TCP SYN flood attacks.

A seminal work of Savage et al. [12] (Probabilistic Packet Marking – PPM) started out research in the field of packet marking for tracing back the source of spoofed packets. These marking methods aim at encoding ID of routers along the path into the packet. Other works extended probabilistic packet marking by authentication and upstream router map [14] or by dynamic marking probability [10]. A similar approach was proposed in [1], but instead of the full path only an address of the interface at the first router is encoded into the packet.

Strayer et al. [15] developed an alternative approach. Rather than to store path in the packets, a router along the path stores information about a packet seen in its Bloom filter. The traceback is performed by querying relevant routers if their local Bloom filter contains the packet.

Other method based on ICMP packets was proposed in [3]. The routers along the path generate with some probability an ICMP packet containing the previous and the next hop of a packet. Such information is sent to the destination which can eventually recover the whole path of the spoofed packets.

Although the previously stated methods are related to our work we do not aim at preventing nor tracing back spoofed packets. Our goal is to detect spoofed packets, identify their destination (potential victim or reflector) and a set of links the spoofed packets are entering the network. The detection of spoofed packets has only been researched in [7, 18, 11]. The first two detection methods are based on detecting variances in TTL (Time To Live). In [18] the authors have discussed TTL issues which constitute a problematic estimation of initial TTL (consider NAT, change of routes, etc.) and a possibility to spoof TTL value. In [11] the authors suggest to detect spoofing periods based on a significant increase of new source IP addresses. Such an algorithm cannot detect spoofing used during reflector attacks since there is only one new spoofed source IP address, i.e., the address of the final victim. We utilize this algorithm as a part of our scheme.

We build our detection scheme on the network processes that are out of the control of a spoofing source. As a result, the scheme is able to work upon NetFlow v5 records. NetFlow v5 is widely spread monitoring protocol supported by routers and other stand alone monitoring probes and exporters [6, 5]. Unlike preventive or tracing methods, our detection method does not require any modification of a packet, no specific protocol nor any modification to the routers. The scheme only assumes that each or the majority of border links of the target network is monitored via NetFlow v5.

3 Detection algorithm

The core idea of the algorithm is to detect the source IP addresses that are not expected to appear in arriving packets on a particular link (entry point in the destination network). The detection is based upon filtering and modeling of

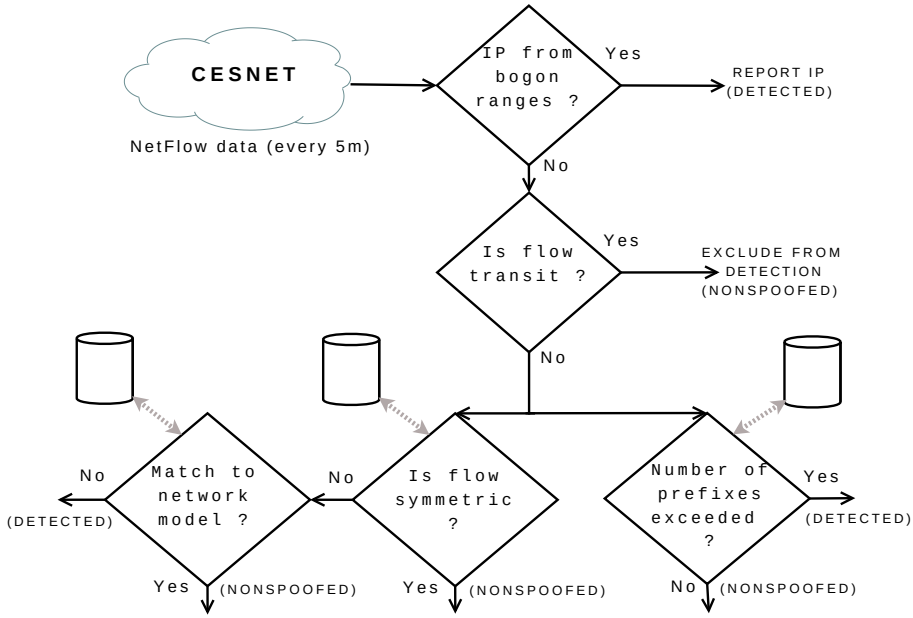

Fig. 1. Algorithm scheme.

the arriving traffic (a new set of flow records is processed periodically every 5 minutes). The proposed scheme is depicted on Figure 1.

The algorithm starts with the filtering of source IP addresses that should never appear at the entry points. These addresses fall either into a set of so called bogon prefixes [17] (e.g., private networks, loopback, etc.) or into a set of prefixes belonging to the destination network itself (in our case approx. 60 prefixes belonging to /16–/24 networks). The intuition is that the randomly spoofed source IP addresses may fall into these prefixes. In such a case, the flow is filtered out and reported as spoofed.

The next step of the algorithm is to reduce the set of loaded flows by removing those which are not important for the detection process. In our case, the transit flows are excluded. These flows traverse via the monitored network but their source and destination addresses belong to other networks. Therefore these flows are out of the detection scope. The transit flow can be recognized easily (the same flow key appears in the incoming as well as in the outgoing traffic) and it is filtered out.

Next, we assume that most of the traffic is transferred over a symmetric path. To this end, the symmetric filter builds a routing model by observing outgoing flows on each link. If the traffic with a particular destination is routed over a particular link it is very likely that the same link is used for the reverse path. The model contains a set of the source IP prefixes derived from the destination IP addresses of the outgoing traffic for each link. It is necessary to process all

Number of entry points	1	2	3	4
Number of source prefixes (/24)	758103	33697	48	1

Table 1. Histogram of the number of links an incoming source prefix is observed on.

Time	Distribution			Average	Received
$[5min]$	e^1 [%]	e^2 [%]	e^3 [%]	a $[flows]$	r $[flows]$
1	100	0	0	1000	1000
2	20	80	0	1750	4000
3	100	0	0	1562	1000
4	0	100	0	1421	1000
5	50	50	0	1316	1000

Table 2. History model for a single prefix containing the distribution for 3 entry points (e^1, e^2, e^3) and the average number of flows (a).

outgoing flows prior to the incoming flows. The incoming flows verified by the model are considered to be legitimate (and are filtered out) whereas the incoming flows taking the asymmetric path may be spoofed and must be classified by the next filter. Also, it may not be sufficient to model expected source prefixes due to a following issue. A path from an external point A to some internal points B and C may differ since these points may be located in different parts of the network. Table 1 shows a histogram of the number of links the source prefixes is observed on. It may be observed that the majority of prefixes occur only on a single link which offers the filter a potential for successful filtering based only on the source prefixes.

Further, the flows are classified by the history-based filter. The filter builds a model of arriving source prefixes (we discuss the length of the prefix in Section 4). There are several issues that must be taken into account when the filter utilizes the model to filter out the spoofed flows. It must deal with load balancing (the source prefix may occur on multiple links at the same time) and route flapping (the link for the source prefix may change frequently). These issues are addressed by the parameters and the characteristics of the model. For each source prefix, the model stores the distribution among the links, exponentially weighted moving average of the flows belonging to each prefix and the time of the last update. Table 2 depicts an example of a single record (columns 2 – 4) for a sequence of intervals. The number of received flows in the current interval is represented by the last column.

The spoofing is detected if there is a deviance from both characteristics — a change of the distribution and a large increase of the received flows (as depicted in the second row in Table 2). A logical expression 1 describes the detection.

$$a_t > ka_{t-1} \wedge \sum_{i=0}^{L} |e_t^i - e_{t-1}^i|/L > H, \tag{1}$$

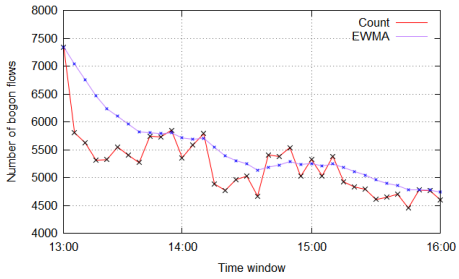

Fig. 2. Number of flows detected by the bogon prefix filter.

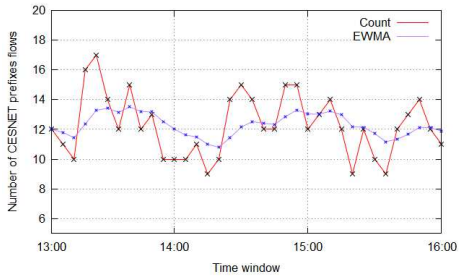

Fig. 3. Number of flows detected by the CESNET filter.

where a_t corresponds to the average number of flows at time interval t and e_t^i is the distribution of the prefix over the links $i = 0 \ldots L$. The coefficient k corresponds to the increased ratio of the average number of flows whereas the threshold H corresponds to the average per-link distribution deviance and L is the number of links. If the spoofing is detected the matching prefix and the link with the largest increase in received flows is reported.

The last model is based on [11]. It receives all flows entering the network except the transit flows. The model tracks the number of received flows per each destination (CESNET) prefix. We follow the implementation in [11] and use CUSUM to detect the increased number of new source prefixes. The detector triggers an alert if the threshold is reached and the triggered destination prefix is reported.

4 Evaluation

The monitoring data all originate from the CESNET network. CESNET is connected to other AS with seven links. All these seven links are monitored and reported NetFlow data are processed. In this study we utilize data from period 11.3.2013 00:00 to 17.3.2013 23:59. Each 5 minute interval contains approximately 19 million of flows collected from all entry points. There are no DoS or DDoS attacks reported by our analysis tools nor by Warden[3] in the data set.

The detection results of the first filter (detecting bogon and CESNET prefixes) are depicted in Figure 2 and 3. Figures 2 and 3 show the occurrence of spoofed addresses from the specific ranges in the inbound traffic. The large number of detected bogons confirms the lack of ingress filtering in other networks. The most often reported prefixes belong to the private network ranges. The spoofing is detected if the observed value differs from the average (EWMA) three times the standard deviation or if the value exceeds a fixed threshold of 13000 flows.

[3] Early warning system deployed by the connected institutions in the stub networks

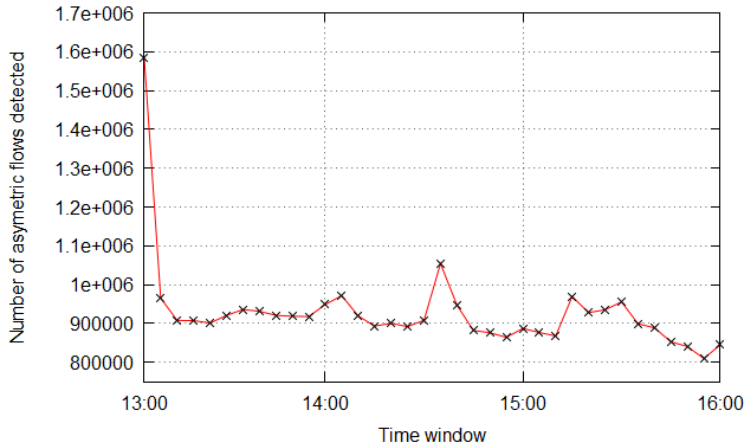

Fig. 4. Number of flows detected by the symmetric filter as potentially spoofed.

Prefix length	/8	/16	/24	/32
Records in model	4 556	227 074	1 442 638	2 420 396

Table 3. Network model size vs. prefix lengths based on a single 5-minute interval.

There are rare cases when the CESNET prefixes appear in the source addresses in the arriving packets. The investigation of these cases revealed misconfigurations of external routers which upon arrival of particular packets return them to its source by the default path. Due to a low amount of these anomalies the detection threshold of CESNET prefixes is relatively low. We set the fixed threshold to be thirty detected spoofed addresses which is two times more than the maximum observed value per 5-minute interval in a long term (a week). Hence the spoofing detector tolerates these anomalies.

Subsequently, the transit filter reduces the processed set of the flows by 25% on average.

The results of the symmetric filter are presented in Figure 4. The model reaches its stable state after it overcomes a learning phase during the first several intervals. A large portion of symmetric communication in our traffic allows to filter out large number (approx. 85%) of flows as legitimate. The number of records stored in the model is dependent on the prefix length (see Table 3).

In all our experiments we utilize prefix /24 to achieve moderate memory requirements and low processing overhead. Additionally, the number of records in the symmetric as well as in the history-based model is dependent on the length of considered history (see Fig. 5). We keep all records that are no older than 60 minutes. It can be observed that stabilizes after initial growth and decrease. The decrease is caused by an increased number of new flows arriving in the first interval due to the symmetric filter which has not built the model yet.

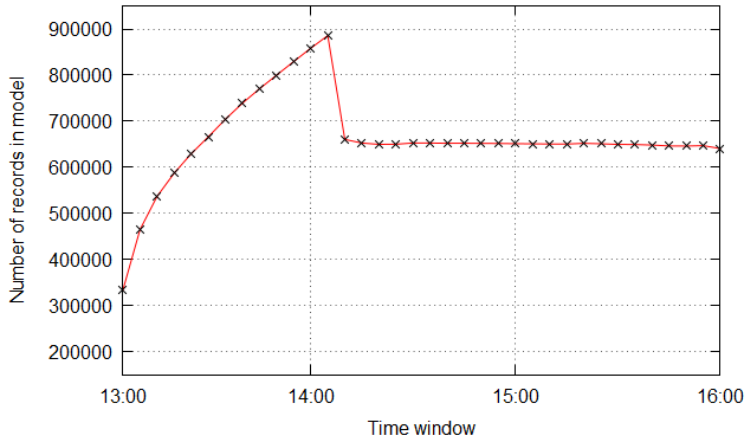

Fig. 5. Network model size vs. length of kept history.

The number of asymmetric flows remains too high for any manual inspection. To this end, the history-based filter matches the flows to the derived model of the arriving source IP prefixes. Based on the observation of the collected data we set up the detection thresholds $H = 40\%$ and $k = 3$. Moreover we introduce an activity threshold of the prefix. This threshold disables matching of the traffic against the prefixes that are not mature enough to be trusted. We have found out that if we set up the activity threshold to be even 5 minutes the number of false positives decreases to zero. Such a setup is aligned with the standard behavior of our network traffic. Figure 6 depicts the situation when the activity threshold is not utilized. After a learning phase in the first few intervals the detector reports only a small number of source prefixes. The second peak is caused by the expiration of the prefixes learned during the first interval when the symmetric filter did not filter out large portion of otherwise symmetrical flows.

At last, we evaluate the model of new source IP prefixes. The detector runs in parallel to the symmetric and the history-based detectors. The number of new source IP prefixes per selected destination prefixes is depicted in Figure 7. Of course, the number of the new prefixes decreases with each new interval as the model learns from new addresses. The average value stabilizes at approx. 130 000 across all destination prefixes in total. The learned prefix is removed from the model after a week of inactivity. The CUSUM detects the increase of the new prefixes with respect to the average value. The outcome of the detector is binary — either the destination prefix receives significant number of new prefixes addresses or not.

The filter detectors are connected in a pipeline in order to reduce the number of flows that must be inspected by the history-based filter. Although the number of alerts is relatively small we do not expect an operator to inspect the

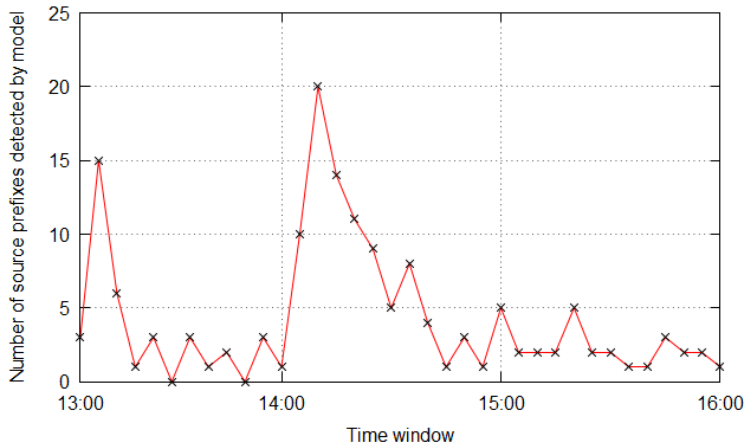

Fig. 6. Number of source prefixes reported by the history-based filter.

corresponding flows. Rather, we envision that the flows are inspected only in the cases when more than one detector agree upon that the interval contains flows with spoofed IP addresses.

5 Conclusion

The paper proposed a new scheme to detect source IP address spoofing. The scheme detects spoofing by analysing NetFlow data collected at the entry points in the network. The scheme is based on four assumptions related to the symptoms of IP spoofing in the network traffic. An offline evaluation of the scheme was done on real data collected from CESNET2 network. The results showed the effectiveness of each assumption. The experiments with parameters of the algorithm revealed the behavior of the detection scheme and provided a hint on setting up the scheme in other networks.

Our future research will focus on proposing further filters to improve accuracy of the whole scheme. For example, if we used IPFIX protocol as an input data, it would be possible to use TTL to create another filter based on [7]. We also work on an NfSen plugin that implements the proposed detection scheme and we plan to run it online.

Acknowledgement

This work was supported by the research programme MSM 0021630528, the grant BUT FIT-S-11-1, the grant VG20102015022 and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

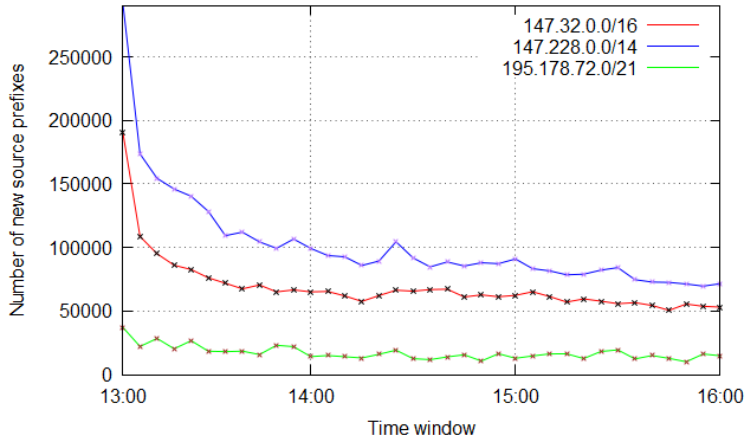

Fig. 7. Number of source prefixes reported per destination prefix by the model of new source prefixes.

References

1. Belenky, A., Ansari, N.: IP traceback with deterministic packet marking. Communications Letters, IEEE 7(4), 162–164 (April 2003)
2. Bremler-barr, A., Levy, H.: Spoofing prevention method. In: Proc. of IEEE INFOCOM (March 2005)
3. Dan, A.M., Usc/isi, D.M., Felix, S., Ucdavis, W., Ucla, L.Z., Wu, C.S.F.: On Design and Evaluation of "Intention-Driven" ICMP Traceback. In: In Proceedings of IEEE ICCCN (Oct 2001)
4. Ferguson, P., Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (May 2000)
5. fprobe: fprobe (March 2011), "http://fprobe.sourceforge.net"
6. INVEA-TECH: Flowmon (March 2011), "http://www.invea-tech.com/products/flowmon"
7. Jin, C., Wang, H., Shin, K.G.: Hop-count filtering: an effective defense against spoofed DDoS traffic. In: Proceedings of ACM CCS 2003 (Oct 2003)
8. Li, J., Mirkovic, J., Ehrenkranz, T., Wang, M., Reiher, P., Zhang, L.: Learning the valid incoming direction of IP packets. Comput. Netw. 52(2), 399–417 (Feb 2008)
9. Lyon, G.F.: Nmap Network Scanning. Insecure, USA (Dec 2008)
10. Peng, T., Leckie, C., Ramamohanarao, K.: Adjusted Probabilistic Packet Marking for IP Traceback. In: NETWORKING 2002. LNCS, vol. 2345, pp. 697–708. Springer-Verlag, London, UK, UK (May 2002)
11. Peng, T., Leckie, C., Ramamohanarao, K.: Detecting distributed denial of service attacks using source ip address monitoring. In: In Proceedings of the Third International IFIP-TC6 Networking Conference. pp. 771–782. Springer (2002)
12. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for IP traceback. SIGCOMM Comput. Commun. Rev. 30(4), 295–306 (Aug 2000)
13. Shen, Y., Bi, J., Wu, J., Liu, Q.: A two-level source address spoofing prevention based on automatic signature and verification mechanism. In: Computers and Communications, 2008. ISCC 2008. pp. 392 –397 (July 2008)

14. Song, D.X., Perrig, A.: Advanced and authenticated marking schemes for IP traceback. In: Proceedings of INFOCOM 2001. vol. 2 (April 2001)
15. Strayer, W.T., Jones, C.E., Tchakountio, F., Hain, R.R.: SPIE-IPv6: Single IPv6 Packet Traceback. In: Proceedings of LCN '04. Washington, DC, USA (Nov 2004)
16. Systems, C.: NetFlow Services Solutions Guide (July 2007), "http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products_implementation_design_guide09186a00800d6a11.html#wp1030098"
17. Team Cymru Inc.: The bogon reference (April 2012), "http://www.team-cymru.org/Services/Bogons/"
18. Wang, H., Jin, C., Shin, K.G.: Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Trans. Netw. 15(1) (Feb 2007)
19. Wanner, R.: Session Hijacking in Windows Networks. SANS Inst. (Oct 2006), "http://www.sans.org/reading_room/whitepapers/windows/session-hijacking-windows-networks_2124"
20. Xie, L., Bi, J., Wu, J.: An Authentication Based Source Address Spoofing Prevention Method Deployed in IPv6 Edge Network. In: Proceedings of ICCS 2007. Springer-Verlag (May 2007)
21. Zuquete, A.: Improving the functionality of SYN cookies. In: In Proc. of IFIP TC6/TC11 Communications and Multimedia Security. pp. 57–77 (Sep 2002)