Comparative Evaluation of Spoofing Defenses

Jelena Mirkovic *Member, IEEE,* and Ezra Kissel

***Abstract*—IP spoofing exacerbates many security threats, and reducing it would greatly enhance Internet security. Seven defenses that filter spoofed traffic have been proposed to date; three are designed for end-network deployment, while four assume some collaboration with core routers for packet marking or filtering. Because each defense has been evaluated in a unique setting, the following important questions remain unanswered: (1) Can end networks effectively protect themselves or is core support necessary? (2) Which defense performs best assuming sparse deployment? (3) How to select core participants to achieve best protection with fewest deployment points?**

This paper answers the above questions by: (1) Formalizing the problem of spoofed traffic filtering and defining novel effectiveness measures, (2) Observing each defense as *selfish* (it helps its participants) or *altruistic* (it helps everyone) and differentiating their performance goals, (3) Defining optimal core deployment points for defenses that need core support, and (4) Evaluating all defenses in a common and realistic setting. Our results offer a valuable insight into advantages and limitations of the proposed defenses, and uncover the relationship between any spoofing defense's performance and the Internet's topology.

I. Introduction

IP spoofing has been used in distributed denial-of-service (DDoS) attacks and intrusions. It is also necessary for *reflector* DDoS attacks, where servers reply to spoofed requests and these replies overwhelm the victim whose address was misused.

A. Spoofing Is an Open Problem:

Some researchers believe that spoofing is not an open problem based on: (1) the Spoofer project's study [1] that estimates that 80% of networks deploy ingress filtering and (2) prevalence of non-spoofed DDoS attacks. We now argue to the contrary.

In the Spoofer project [1] distributed volunteers download software that spoofs packets to a monitoring machine. From packet losses, the authors infer the existence of ingress filtering [2] in the volunteer's networks, which drops outgoing traffic carrying addresses not assigned to the deploying network. Spoofer measurements show that around 80% of networks participating in the project deploy ingress filtering. Because the total number of participating hosts is in the low thousands, these results cannot be readily extrapolated to the entire Internet. Further, even if only 20% of all networks allowed spoofing, they could still generate unlimited spoofed and reflected traffic toward any target.

Many contemporary DDoS attacks send valid application requests, and do not use spoofing, but a large number still do. The analysis of backscatter traffic [3] inferred that there were several hundred DDoS attacks with spoofing per day. Another popular trend today is the use of reflectors for recursive DNS attacks [4], which mandates spoofing.

B. Our Focus:

Many approaches have been proposed to handle spoofing during specific attacks, or to trace back sources of spoofed traffic. In this study we focus only on approaches that work in a *generic*, *single-step*, *packet-filter* manner, as we explain next. These approaches associate each IP address with some parameter (e.g., a route to the filter, a secret mark, etc.) via a parameter table. When a packet arrives, the chosen parameter's value is inferred from it, and compared to the value in the parameter table. Mismatching packets are considered spoofed.

These approaches are *generic* because their only goal is to filter spoofed packets, regardless of the security threat that generated them. They are *single-step* because there is no interactive communication with an alleged packet source when a suspicious packet is received. Finally, these approaches work in a *packet-filter* manner – the parameter table can be visualized as a set of firewall rules that specify allowed traffic, and the default deny rule.

To date, seven approaches have been proposed that fit our scope. **Ingress filtering (ING)** [2] associates all addresses in the deploying network with the outgoing direction of packets, and the rest of the addresses with the incoming direction. A **hop-count filter (HCF)** [5] associates a source with a router hop count between it and the filter. A **route-based filter (RBF)** [6] associates a source with the previous hop traversed by this source's packets. An **inter-domain packet filter (IDPF)** [7] associates a source with the set of *feasible* previous hops that could carry its traffic. **SPM** defense [8] at the traffic's destination associates a source autonomous system (AS) with a secret it exchanged with the defense. The source marks packets with this secret. **Packet passports (PASS)** [9] are attached by participating senders to their packets, and contain a sequence of marks, each derived from a secret shared between the source and one AS on the path to the destination. These marks are associated with the source. **PiIP** [10] deploys distributed *markers* – routers that mark forwarded packets by shifting their IP identification field and appending a short label derived from the previous hop's and this router's IPs. Destinations extract the mark sequence and associate it with the packet's source.

These defenses were evaluated by their authors using custom performance measures and in a customized setting, which hinders comparison. The following questions remain: (1) Which defenses perform well in *isolated, single-filter* deployment, realistic for early adopters? (2) Which defenses perform well in *sparse, random* deployment, realistic for spontaneous adoption? (3) If we could choose a *small number of best deployment points* for each defense, what would the resulting performance be? (4) What is the optimal deployment strategy that yields best protection with fewest deployment points?

Questions (1) and (2) are important because isolated deployment and random, sparse deployment are common in the Internet – each network is driven by its own economic interests to deploy some security measure. If a defense does not offer good protection under these assumptions it is unrealistic and should not be pursued. If no defense offers good protection in spontaneous (isolated or random, sparse) deployment, the next possible strategy is to investigate an Internet-wide deployment that should help everyone, and this is the focus of questions (3) and (4). If a defense performs poorly in an optimal deployment on a small number of well-chosen networks, it is useless and should not be pursued.

C. Contributions and Overview:

This paper aims to answer the questions posed above. The value of our work lies in identifying factors that influence a spoofing defense's effectiveness and specifying how to control these factors to obtain the best protection. While we do not propose any novel defense, our results should advance spoofing research by: (1) Highlighting the fact that a defense's deployment strategy has a paramount impact on its effectiveness. (2) Evaluating existing defenses in a common setting, which clearly shows which approaches hold promise for practical deployment, and which are inferior. The detailed contributions of this paper are:

(1) We define a theoretical framework for performance analysis of spoofing defenses (Section II). We show that any defense's success depends on two key factors: the placement of packet filters on well-traversed routes, so they can intercept spoofed traffic (filter *popularity*), and the ability of filters to restrict spoofing in intercepted packets (filter *strength*). Filter popularity depends on the Internet topology and routing. Filter strength depends on the Internet topology for RBF, IDPF and PiIP – better connected filters have a higher impact. For ING, SPM and PASS filter strength depends on the size of the deploying networks' address spaces, while the strength of HCF filters is always high regardless of their deployment. These observations guide our strategy for selection of optimal deployment points.

(2) We define three intuitive defense performance measures (Section III) that express the reduction of: (a) spoofed traffic reaching its destination, (b) addresses that can be used in reflector attacks, (c) number of attack locations that can spoof at will.

(3) We evaluate the effectiveness of all filtering defenses proposed to date in a common setting (Sections IV and V), replicating Internet topology and routing at the autonomous system level. Our results indicate that three defenses (HCF, RBF and PiIP) would bring significant spoofing reduction to all Internet users, and across all dimensions, if deployed at 50 selected autonomous systems owned by 18 tier-1 ISPs. Isolated and sparse, random deployment can only protect deploying networks against spoofed traffic, but not against reflector attacks. This protection is moderate to poor for all defenses but HCF. We also briefly discuss other relevant defense features, such as cost, security and likelihood of decision errors.

Since filter popularity and some filter strenghts depend on the Internet topology and routing, it appears at first blush that the accuracy of their models will critically influence evaluation outcomes. We discuss several sources of this data in Section IV-A and argue for the approach we adopted. In Section V we show that the defenses' effectiveness depends mostly on the existence of highly popular and large Tier-1 nodes, which are present in topologies inferred from different sources. In Section VI, we discuss and evaluate the impact that alternative topologies and routing approaches would have on our results. We conclude in Section VII.

II. Analysis of Defense Effectiveness

Let IP_{rout} and *IPv4* be the set of globally routable and all IP addresses, respectively. During the analysis, we observe the Internet as a directed, connected graph whose nodes are routers or autonomous systems, and whose links are determined by routing protocols. We consider packets sent from source address $s \in IP_{rout}$ to destination address $d \in IP_{rout}, d \neq s$, spoofing the address $p \in IPv4$, $p \neq s$. In the analysis, we investigate factors that determine the portion of possible $\{s, d, p\}$ combinations filtered by some defense.

A. Single Filter

Assume that some spoofing defense is deployed only at a node F. For each source/destination pair $\{s, d\}$ we define the mapping $hit_F(s, d)$, to be 1 if the path from s to d contains F, and 0 otherwise. All approaches of interest detect spoofed packets by building a table that associates source addresses (aggregated at some granularity) with some parameter as summarized in Table I. Mapping of sources to parameters is frequently many-to-one, due to aggregation of source addresses in the table or due to sharing of paths between sources that results in sharing of parameter values. Thus, F will be able to detect spoofed packets only for some s and p combinations, when the parameter values associated with these addresses are different. We express this through the mapping $diff_F(s, p)$, which is 1 if F can detect s spoofing p, and 0 otherwise.

TABLE I
Parameter associated with a source IP

Defense	Parameter
ING	Traffic direction.
HCF	Hop count.
RBF	One previous hop.
IDPF	Set of feasible previous hops.
SPM	Packet mark, placed by sender. Mark is destination dependent, route and packet independent.
PASS	Sequence of packet marks, placed by sender. Each mark is destination, route and packet dependent
PiIP	Sequence of packet marks, each router places one fixed mark.

A packet from s to d, spoofing p, will be filtered out by F if and only if the packet hits F and F can distinguish between s and p, that is only if both $hit_F(s, d) = 1$ and $diff_F(s, p) = 1$. We define the ***filtering function***: $filter_F(s, d, p) = hit_F(s, d) \cdot diff_F(s, p)$, and we define the ***filter impact*** of F as the number of all possible $\{s, d, p\}$ combinations that are filtered by F:

$$impact_F = \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} \sum_{p \in IPv4} filter_F(s, d, p) \qquad (1)$$

We define the ***filter strength per source*** s as the number of p values that packets from s cannot spoof if they hit this filter: $strength_F(s) = \sum_{p \in IPv4} diff_F(s,p)$, and we define the ***filter strength*** as the aggregate of filter strength per source for all sources: $strength_F = \sum_{s \in IP_{rout}} strength_F(s)$. Similarly we define the ***filter popularity per source*** s as the number of destinations d such that paths from s to d cross this filter: $pop_F(s) = \sum_{d \in IP_{rout}} hit_F(s,d)$, and we define the ***filter popularity*** as the aggregate of filter popularity per source for all sources: $pop_F = \sum_{s \in IP_{rout}} pop_F(s)$.

We can express the impact of a filter as a composition of its popularity and strength at the source level:

$$impact_F = \sum_{s \in IP_{rout}} pop_F(s) \cdot strength_F(s). \quad (2)$$

Thus both popularity and strength play an important role in defining a filter's impact, and interact at the single source granularity. To have a high impact, a filter need not only be popular and strong, but must be *popular and strong for the same sources*.

B. Multiple Filters

We now assume that a set of N filters $FS = F_1 \dots F_N$ is deployed and investigate the collective impact of this filtering. The ***joint filtering function*** is:

$$filter_{FS}(s,d,p) = \bigvee_{F \in FS} filter_F(s,d,p) = \bigvee_{F \in FIL(s,d)} diff_F(s,p), \quad (3)$$

where $\bigvee$ denotes a logical or operation and the mapping $FIL(s,d)$ returns the set of filters traversed by traffic from s to d. Eq. (3) says that a packet from s to d, spoofing p, will be filtered if it hits at least one filter that can distinguish between s and p. The ***joint filter impact*** of FS is:

$$impact_{FS} = \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} \sum_{p \in IPv4} filter_{FS}(s,d,p) \quad (4)$$

$$= \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} \sum_{p \in IPv4} \bigvee_{F \in FIL(s,d)} diff_F(s,p) \quad (5)$$

$$= \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} \left| \bigcup_{F \in FIL(s,d)} \{p | diff_F(s,p) = 1\} \right| \quad (6)$$

For some filter set X we define the ***joint filter strength per source*** s as:

$$strength_X(s) = \left| \bigcup_{F \in X} \{p | diff_F(s,p) = 1\} \right| \quad (7)$$

The impact can then be expressed as the filter strength of set $FIL(s,d)$ per source, aggregated across all sources s and destinations d:

$$impact_{FS} = \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} strength_{FIL(s,d)}(s) \quad (8)$$

The joint filter impact again depends on the joint filter popularity hidden in $FIL(s,d)$, which we call ***path coverage***, and on the joint strength of filters on diverse paths. To maximize the impact of N filters we need to select deployment points that lie on many paths, have the ability to detect and discard many $\{s, p\}$ combinations, and the combinations on each path overlap minimally (to maximize the union in Eq. 7).

III. Defense Performance Measures

There are three dimensions of spoofing: spoofed addresses (p), sources of spoofed traffic (s) and its targets (d). The main goal of a spoofing defense is to provide protection to targets against spoofed and reflected traffic. We express this through the *target protection* and *reflector attack protection* measures, respectively.

When we evaluate these measures we will assume that the remaining two dimensions – $\{s, p\}$ in case of target protection and $\{s, d\}$ in case of reflector attack protection – are distributed uniformly at random in the IPv4 space. We do this because we cannot predict which addresses may be spoofed and towards which targets. The observed distribution of Internet attackers is not uniform, with attackers showing strong preference towards a few networks that are poorly secured [11], [12]. We cannot assume such distribution in our evaluation because: (1) There is no public information about exact locations that attackers prefer, so at best we could make a random guess. Since defense's performance depends not only on attacker distribution but on their exact locations, this would lead to wrong results. (2) It is known that popular attacker locations change over time [11], [12] and in an unpredictable manner, so our results would quickly become outdated. Since no one can exactly predict likely placement for future attackers, it is fair to assume, as we do, that any network is equally likely to host them. Our measures express protection offered to any victim in this scenario. Further, we evaluate how lucrative attack locations are after a defense is deployed via the *attacker impairment* measure.

A. Target Protection Measure

Target protection (*TP*) measure for node x defines the number of $\{s, p\}$ combinations that will be filtered en route to destination x. It expresses protection of node x from spoofed traffic, assuming random deployment of attacking machines and random spoofing.

$$TP(x) = \sum_{s \in IP_{rout}} \sum_{p \in IPv4} filter(s,x,p) = \sum_{s \in IP_{rout}} strength_{FIL(s,x)}(s)$$

$TP(x)$ depends on the number of filters hit by traffic from various sources to x, and the filter strengths. For many defenses, *TP* measure for filter-deploying networks will be higher than for legacy networks because all spoofed traffic sent to a filter-deploying network hits at least one filter.

B. Reflector Attack Protection Measure

Reflector attack protection (*RAP*) measure for node x defines the number of $\{s, d\}$ paths on which packets spoofing x will be filtered out. RAP measure expresses protection of node x from reflected traffic assuming random selection of attack sources and destinations. This protection is not achieved directly by filtering reflected traffic, but indirectly by filtering spoofed service requests and thus preventing reflected traffic.

$$RAP(x) = \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} filter(s,d,x) = \sum_{s \in IP_{rout}} \sum_{d \in IP_{rout}} \bigvee_{F \in FIL(s,d)} diff_F(s,x)$$

RAP(x) depends on the path coverage (*FIL*(s, d)) and the filters' ability to detect spoofing of the address x. We will show that isolated defenses and collaborative defenses that are deployed randomly and sparsely cannot provide good protection against reflector attacks, because they do not have sufficient path coverage.

C. Attacker Impairment Measure

Attacker impairment (*AI*) measure for node x defines the number of $\{d, p\}$ combinations in spoofed traffic generated by x that will be filtered. It expresses the impairment of node x's spoofing ability, if it were recruited as a bot.

$$AI(x)=\sum_{d\in IP_{rout}}\sum_{p\in IPv4} filter(x,d,p)=\sum_{d\in IP_{rout}}\sum_{p\in IPv4}\bigvee_{F\in FIL(x,d)} diff_F(x,p)$$

AI(x) will be high if x's traffic crosses a large number of filters that can distinguish x from many addresses. Nodes that reside in vicinity of filters should be less lucrative for attackers since there is a good chance that many of their routes cross a filter. We will call a node *very impaired* if *AI*$(x) >= 0.95$ and *moderately impaired* if $0.95 > AI(x) >= 0.90$.

D. Cost and Security

In addition to a defense's effectiveness, factors that influence its suitability for real-world deployment are cost and potential security vulnerabilities.

Since the implementation of many spoofing defenses is not open-source, we could not measure their cost directly. Instead, we infer a defense's cost from its design, and focus only on per-packet processing and storage costs. Because the parameter table update is still an open problem for some defenses (e.g., RBF, IDPF), we cannot include parameter table setup and update costs in our calculation.

TABLE II
ELEMENTARY COST COMPONENTS

Category	Variable	Component Description	Cost
Per-packet	l	Table lookup and comparison with some packet value	8 ns
	m_d	Insert a deterministic mark into packet	8 ns
	m_c	Insert a cryptographic mark into packet	1 μs
	v_c	Verify a cryptographic mark	250 ns
Storage	N_{AS}	Number of ASes in the Internet	30 K
	N_T	Number of (adequately aggregated) entries in the parameter table	200 K
	N_l	Number of links at an AS	280
	N_p	Number of prefixes in an AS	5
	N_{ASh}	Number of AS hops from source to destination	4

For each cost category we define the elementary cost components (Table II) and express the category's cost as their combination. Table lookup and placing a deterministic mark in the packet can usually be done at forwarding speed. We estimate the maximum cost of these operations (separately and together) conservatively, by dividing the highest router speed today – 40 Gbit/s – by the smallest packet size of 40 B for IP and transport headers. The costs of cryptographic marking and verification are taken from [9]. We calculate the number of ASes from our topology inferred as in [13], we take the average value from the same topology for the number of prefixes per AS and number of AS hops between a source and a destination, and we take the average value for well-connected ASes (> 100 neighbors) for the number of links per AS. Those well-connected ASes are the best candidates for filter placement as we will show in the evaluation. Finally, we estimate the number of entries in the parameter table from the number of routing entries in BGP tables in today's core routers, as given in [14].

Similarly to defense cost, we could not assess security vulnerabilities of proposed defenses through experimentation but through their design. We focus on attacks that create spoofed packets that bypass a given defense and differentiate between several such approaches that are shown in Table III. We assume that all such attacks are created by end hosts while routers deploying defenses are trustworthy.

TABLE III
SPOOFING APPROACHES THAT BYPASS A GIVEN DEFENSE

Power	Name	Description
6	spoof-all	An attacker's spoofing ability is only slightly affected by the chosen defense
5	possible-path-all	An attacker that lies on an advertised but unused route between a source and a filter can spoof this source to any destination
4	path-all	An attacker that shares path between a source and a filter can spoof this source to any destination
4	subnet-all	An attacker that shares a subnet with a source can spoof this source to any destination
3	sniff-all	A sniffing attacker learns some parameter value that enables him to spoof the source of the sniffed packet to any destination
3	replay-all	A sniffing attacker can replay sniffed packets to any destination
2	replay-fix	A sniffing attacker can replay sniffed packets only to the destination in the packet
1	probe-fix	A probing attacker learns some parameter value that enables him to spoof the source for which the probing was done to the destination that was probed

The first column in table III shows the attack power. This is simply a rank that denotes how desirable and difficult this attack is from an attacker's perspective, when compared to other attacks. Note that some defense's vulnerability to a specific attack is a feature of the *current* defense design – it is not necessarily inherent and sometimes can be countered by additional mechanisms.

E. False Positives and False Negatives

Another important measure of a defense's effectiveness are frequencies of false decisions: failures to detect spoofed packets (false negatives) and false detections of legitimate packets as spoofed (false positives). Filtering defenses we explore in this paper base all decisions on their parameter table. False positives occur only if data in this table is wrong or stale, and are discussed in Section V. False negatives occur if a spoofed packet matches the parameter value from the table for the address it spoofs or if a spoofed packet bypasses all filters that could detect it. A successful match can happen because: (1) an attacker's machine shares the same parameter value with the spoofed address, (2) an attacker made a lucky guess

or (3) an attacker learned proper parameter values via probing or sniffing. False negatives due to parameter value sharing or filter bypass can be calculated as $100\% - max(RAP, TP)$ for a given number of filters. We discuss the possibility of cases (2) and (3) in Section V.

IV. Defense Evaluation Method

We evaluate effectiveness of the proposed defenses by first reproducing the Internet's autonomous system (AS) map using the connectivity and AS relationship information inferred for May 2005 via the approach described in [13]. We then use No-Valley-Customer-Prefer approach [15] to infer routing behavior from AS relationships. During evaluation we calculate parameter tables for each defense, generate packets that traverse $\{s, d, p\}$ parameter space, and calculate performance measures defined in Section III. We now first discuss different inference approaches for the AS connectivity, relationship and routing, and we provide arguments for the approach adopted in this paper. We then explain how the evaluation is performed and how we specify performance goals for each defense.

A. AS Topology and Routing Inference

Ideally, all our evaluations should be done on the router-level topology of the entire Internet. Since such a topology is not available, we resort to evaluation on the AS-level topology. The following properties of the Internet topology must be faithfully reproduced to guarantee the correctness of our evaluation: (1) Inter-AS routing and (2) AS address space size. The AS address space size can be easily inferred from the RouteViews data [16] by assigning the size of each unique prefix to the AS, which is the last hop on the route to this prefix. On the other hand, inference of inter-AS routing is difficult due to a lack of necessary information, such as a global database of the routing tables.

One commonly used approach is to infer AS connectivity, and then apply shortest-path routing on this graph [6], [17], but this is unrealistic because Internet routing is not shortest-path. Another approach, which we chose, is to infer AS relationships in addition to connectivity and use this information to set up routing by applying No-Valley-Customer-Prefer principle [15] In reality, this principle is used in conjunction with private BGP policies to make routing decisions, so inferred routes may still differ from real ones, but they will be more accurate than if we used the shortest-path approach. Finally, in [18] Mulbauher et al use iterative learning and multiple quasi routers per AS to learn routing policies from a large set of private and public routing data, and use them to infer realistic routes. Because their data is not public, we could not use it in our study. But we obtained from the authors of [18] routing tables for a 1,000-node subset of the AS topology. This results in a $\approx 10,000$ node topology (inferred from routes), which we call the "MP topology," with complete routing information for 1,000 ASes. We use this small topology in Section VI to evaluate the impact of multipath forwarding on our results.

Several Internet measurement projects have produced information that can be used to infer AS-level topology, AS-relationships, or both. To date, there is no agreement among researchers which source is the most complete. We investigated RouteViews [16], WHOIS [19], Skitter [20], NetDimes [21], IRL UCLA [22] and UCR [13]. Among these sources, only RouteViews [16] and UCR [13] contain data needed to infer AS relationships in addition to connectivity. Since we need the AS relationship data to infer routing and also to populate IDPF's parameter tables, we are limited to these sources. We decided to use the UCR source since it provides more detailed and accurate AS connectivity and relationship information. The UCR data is extracted by a large-scale comprehensive synthesis of publicly available information sources such as BGP routing tables, Internet Routing Registries, traceroute data, and Internet Exchange Points (IXPs). Data extracted from IRR is filtered by the Nemecis tool [23], to remove inconsistent, incorrect, or obsolete routes. Data is extracted from IXPs using state of the art mechanisms to identify potential edges. Ultimately, every edge in the graph is confirmed either by a BGP table or a traceroute. We call this inferred topology the "UCR topology."

While the UCR topology is the most complete to date, it may still be far from the ground truth. He et al. in [13] admit that their method of link inference may miss between half and 80% of peer-to-peer (P2P) links. Independently, Oliveira et al. [24] find that public link sources miss almost no provider-customer links and Tier-1 links, but may miss between 80% and 95% P2P links, mostly between lower-end ISPs, content providers and stub nets. We examine the effect of missing links in Section VI and show that the addition of more P2P links has a minor impact on our evaluation results and conclusions.

B. Evaluation Methodology

We deploy filters (and populate parameter tables) on the AS map at the AS level, following some chosen deployment strategy. We then generate all $\{s,\ d,\ p\}$ combinations, i.e. we let all sources send traffic to all destinations, with each source spoofing all other sources. We aggregate this traffic generation at the AS level in each dimension (s, d and p) to reduce computational load. A packet traverses the path from s to d using routing information, and may be filtered by filters on the route. We collect packet drop statistics and convert them into effectiveness measures. AS-level filter deployment allows for *intra-AS spoofing*, when s and d belong to the same AS, because such packets are not seen by filters. It also allows for *own-AS spoofing*, when s and p belong to the same AS, because parameter tables are at the AS granularity. Intra-AS and own-AS spoofing is not part of our effectiveness calculation.

We make the following assumptions during evaluation: (1) Each AS is approximated by a single router, so all its sources follow a single route to a given destination. In reality, 75% of ASes have multiple routes per prefix as shown in [18]. But since we lack information to infer correct multipath routes and correct forwarding policies for each AS, we resort to single-router-per-AS assumption for our UCR topology. In Section VI we show the effect that the multipath forwarding has on our results, by using the MP topology. (2) Any node, filter or not, can detect and filter traffic from addresses that are not globally routable. This filtering is not part of the effectiveness

calculation because the non-routable space is large – it would introduce bias by decreasing differences between performance measures. We limit spoofable addresses to $\{p \mid p \in IP_{rout}\}$. (3) Nodes that deploy any spoofing defense except ingress filtering, also deploy the modified ingress filtering as described in Section V-A. Such filtering is cheap compared to other defenses' cost, and it improves the overall protection.

C. Related Work

For space reasons, we only survey work related to evaluation of spoofing defense effectiveness. There are several papers that address the issue of building parameter tables, or propose a defense very similar to our surveyed defenses in concept but different in table setup or update. These approaches are out of scope for our paper.

Ours is the first work that evaluates several spoofing defenses in a common setting. Our evaluation approach is similar to those used in [6] to evaluate the RBF defense and in [7] to evaluate the IDPF defense. They also build the AS-level topology from the public routing data (Routeviews [16], so their topology is less complete than ours), deploy filters at the AS level and use the number of possible $\{s,\ d,\ p\}$ combinations to generate effectiveness measures. Significant differences between our approach and [6], [7] are: (1) We observe s, d and p dimensions at the ***IP-level***, unlike [6], [7] that observe them at the ***AS-level***. Because the IP size distribution across ASes is far from uniform, AS-level analysis leads to false, much lower effectiveness results. (2) We define an optimal filter selection strategy for each defense, and evaluate defenses in multiple sparse, ***realistic*** deployment scenarios (1–50 filters), while [6], [7] focus on a single, ***unrealistic*** deployment scenario – the vertex cover of the AS graph – which requires several thousand deployment points. Our results show that sparse, strategic deployment is frequently as good and sometimes even better than vertex cover deployment. (3) We use continuous effectiveness measures (e.g., y% of *hosts* cannot receive more than x% of $\{s,\ p\}$ combinations), that better express filtering benefit than binary measures used in [6], [7] (e.g. y% of *ASes* cannot receive any spoofed traffic).

In [8] authors evaluate the SPM defense on an artificially generated and simplified Internet topology at the ISP level, assuming 10,000 nodes. In [5] authors evaluate the HCF defense in an isolated deployment, so only a single filter's protection from spoofed traffic was evaluated, while we evaluate protection of filters and all Internet participants in multiple scenarios. In [10] authors evaluate PiIP using the Skitter topology [20], and assuming random, wide deployment, while we evaluate with a more realistic topology and in more realistic deployment scenarios.

D. Selfish vs. Altruistic defenses

We differentiate between: *selfish* defenses that are deployed by networks for their own protection, and *altruistic* defenses that are deployed to reduce spoofing for everyone. HCF, SPM and PiIP are selfish defenses. With slight modifications that we propose in the following sections they can become altruistic defenses. Ingress filtering, RBF, PASS and IDPF are altruistic defenses.

When we evaluate a selfish defense, we will focus on the protection enjoyed by each participant in isolated, single-filter deployment, which is a realistic deployment pattern as discussed in Section I. If isolated deployment is impossible, e.g., because a defense is collaborative and needs at least two participants, we will assume sparse deployment at random 10, 100 and 1,000 filters.

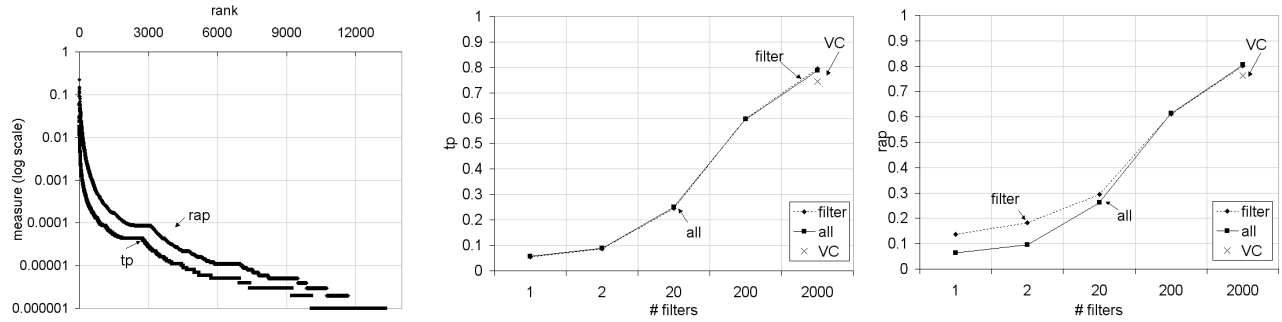
For altruistic defenses we will seek to define an optimal deployment strategy. Such defenses include core nodes, and optimal deployment is realistic since adding new services to core routers is likely to be a community effort, and the deployment should be strategic to minimize cost. We will measure protection enjoyed by participating networks, and by all Internet hosts. Participants' measures, if high, provide the deployment incentive while measures for all hosts are the real performance target for altruistic defenses. We will also measure attacker impairment.

Because vertex cover deployment was proposed for several altruistic defenses [6], [7] we will also evaluate them in this setting for comparison purposes. The vertex cover size is 3,394 nodes in the UCR topology.

E. Optimal Deployment Strategies

Given a fixed cost of N deployment points, an optimal deployment strategy maximizes the number of $\{s,\ d,\ p\}$ combinations that will be filtered, i.e. it maximizes the joint filter effectiveness. This is an instance of the maximum coverage problem, which itself is a variant of the set cover problem, and finding the optimal solution is NP-hard [25]. Instead, we use a simple greedy heuristic which starts with an empty set of deployment points – D and an empty set of filtered $\{s,\ d,\ p\}$ combinations – C. At each step, an AS that can filter the greatest number of combinations not already present in the set C is added to D and the set C is updated.

An additional challenge we faced was the exact implementation of the greedy heuristic. Since the number of $\{s,\ d,\ p\}$ combinations for the Internet, with all possible aggregations, is around $12*10^{12}$ we could not fit required data into memory even if we used a distributed implementation. Instead, our algorithm for optimal filter deployment strategy selects a set of k samples along each of s, d and p dimensions, with the probability of selecting an AS or an address prefix as a sample being proportional to its address size. Our algorithm then finds the N best deployment points that cover combinations of selected samples, applying the greedy heuristic. We repeat this process 10 times, then we rank filters by their selection frequency and choose the most frequent N filters as the optimal deployment points. We found that when $N < 10$, the same filters get chosen in repeated runs, although their order may differ. When $N > 10$, the best 10 filters appear high in selections made in each run, while the remaining filters may appear in some runs but not in others. Choosing the remaining filters from this set based on their frequency, some other criteria or even at random made minor differences in our results.



(a) Isolated deployment: *TP* and *RAP* (b) Optimal deployment: *TP* (c) Optimal deployment: *RAP*

Fig. 1. Performance measures for ING

V. Evaluation Results

We first present results for each defense separately, and then aggregate them into a single table at the end of this section.

A. Ingress Filtering

We consider modified ingress filtering (ING) that removes packets with internal addresses from *both incoming and transit* traffic, and packets with external addresses from outgoing traffic. The addition of transit traffic processing increases the benefits of ingress filtering.

Normalized strength of an ING filter with $|IP_{ing}|$ addresses is: $strength_F = 2 \cdot |IP_{ing}| \cdot |IP_{rout} - IP_{ing}|/|IP_{rout}|^2$. Strong filters are those ASes that own a large address space. Because the AS size follows the power-law distribution, there will be a few strong ING filters.

Figure 1(a) shows *TP* and *RAP* measures for an isolated filter deployment on the y axis, on a log scale. The x axis shows the filter rank. As expected, both *TP* and *RAP* are extremely small for majority of filters. The best *RAP* value is 22% because this AS lies on 22% of source-destination paths and thus cannot be spoofed on these paths. The best *TP* value is 6%. These results confirm that the deploying network does not benefit much from ING.

Fig. 1(b) and 1(c) show *TP* and *RAP* measures for the optimal and vertex cover (VC) altruistic filter deployment for filters and for all Internet hosts, with x-axis showing the number of ASes that are filters on a log scale and y-axis showing the protection measure, also on a log scale. Measures for the VC deployment are shown as an X mark at the rightmost (and usually top) part of the graph; their x coordinate is not to scale to preserve the graph's visibility. Optimal deployment of filters on 2,000 chosen ASes (10% of all ASes) offers 75% protection from spoofed traffic and 77% protection from reflected traffic. This result refutes common belief that ingress filtering has a minor effect unless universally deployed, and comes from the fact that selected filters cover around 80% of the IP address space and jointly have $> 99\%$ popularity. *TP* measures of filters and all nodes are the same because an ingress filter removes the same amount of spoofed traffic from all routes that hit that filter. *RAP* measures are slightly better for filters than for all nodes when deployment is sparse (< 200 nodes) since the joint size of filter address space is smaller than the size of the external address space, but a filter's address cannot be spoofed by external traffic that hits this filter. VC-deployment offers a lower protection than 50 optimal filters, because VC misses some large filters. Observing attacker impairment, 9% of IPs are very impaired and 20% are moderately impaired.

The cost of ingress filtering consists of a per-packet lookup (< 8 ns) and a 25 B storage cost to record internal prefixes, assuming 4 B to record the network prefix and 1 B to record the mask. An attacker located in a network performing ingress filtering can perform subnet spoofing, thus ING is vulnerable to subnet-all, path-all, possible-path-all, replay-all and replay-fix attacks. An outside attacker is only slightly limited, thus ING is vulnerable to spoof-all attack.

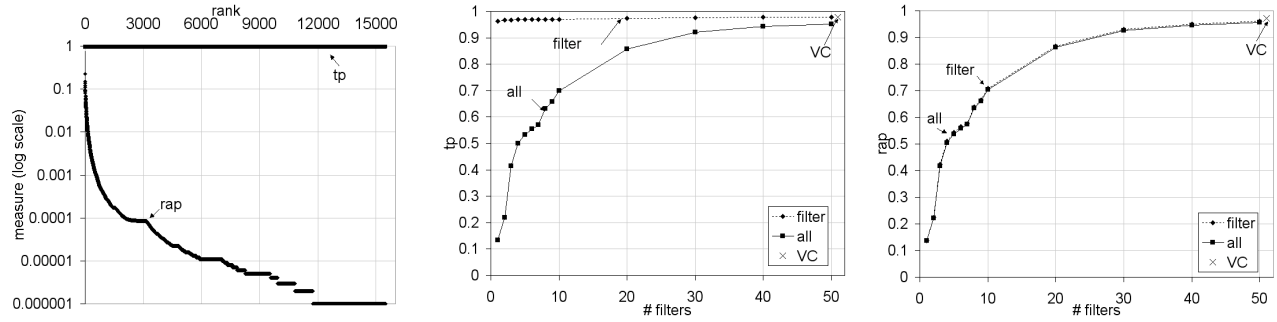
ING's parameter table values change very rarely and are updated manually. Thus the false positive probability is zero. An attacker cannot influence the direction of spoofed packets with regard to filter thus false negatives due to guessing are also zero. Column 1 in Table IV summarizes measures for ING.

B. Hop-Count Filtering

HCF associates each source with the router hop-count between it and the filter. Hop-counts are inferred from the TTLs in packets belonging to established TCP connections. Since we reproduce Internet topology at the AS-level, we mimic router-level hop counts by associating a random hop count chosen from [1–4] inclusively, with each AS-AS link. A packet's hop count at a filter is the sum of the hop counts of traversed AS links. This strategy produces Gaussian hop count distribution, observed in the real Internet [5], and end-to-end hop counts lie within observed limits.

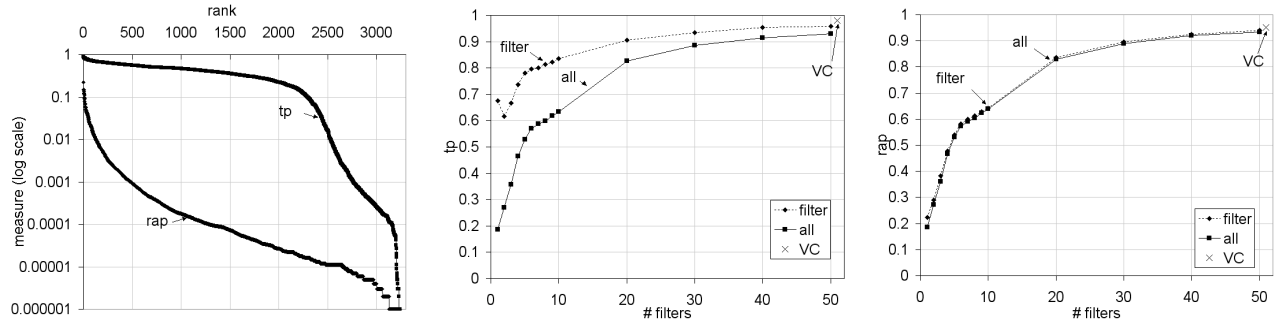
Normalized strength of a hop-count filter is: $strength_F = \sum_{p \in IP_{rout}} Hop(p)/|IP_{rout}|^2$, where $Hop(p)$ is the number of all sources whose hop count differs from p's hop count. Because the node distances on Internet-like graphs, that exhibit power-law distribution of node degrees, follow the Gaussian distribution [26], the strength of HCF filters should be fairly constant and high.

HCF was proposed as a selfish defense. Fig. 2(a) shows the *TP* and *RAP* measures in isolated deployment. The *TP* measure is consistently high, because of high filter strength, making HCF an ideal selfish defense. The *RAP* measure, on the other hand, is low since a single filter does not achieve sufficient path coverage to lower its danger from reflector attacks.



(a) Isolated deployment: *TP* and *RAP* (b) Optimal deployment: *TP* (c) Optimal deployment: *RAP*

Fig. 2. Performance measures for HCF



(a) Isolated deployment: *TP* and *RAP* (b) Optimal deployment: *TP* (c) Optimal deployment: *RAP*

Fig. 3. Performance measures for RBF

HCF can be transformed into an altruistic defense by applying the same filtering approach to the transit traffic. *TP* and *RAP* measures for altruistic deployment are shown in Fig. 2(b) and 2(c), respectively. *TP* and *RAP* measures are very high for optimal deployment and the top 50 HCF filters offer 95% protection to everyone. Filters' *TP* measure is high in sparse deployment offering good deployment incentive, but their *RAP* measure remains low until a sufficient path coverage is achieved. The vertex cover deployment for HCF offers a slightly higher protection (1-3%) than the optimal deployment, but with many more deployment points. Around 4% of IPs are very impaired with regard to hosting attackers, and 71% are moderately impaired. Thus HCF makes around 3/4 of IP addresses unattractive for hosting attackers.

The cost of hop-count filtering consists of per-packet lookup ($<$ 8 ns) and 1.2 MB of storage to record hop count for all source prefixes, assuming that we need 5 B to record the prefix and 1 B for hop count value. A probing attacker can learn the correct TTL value for a given source address, and a given destination [5]. Thus HCF is vulnerable to the probe-fix attack. An attacker located on the path of the traffic can replay all packets to their original destination – replaying them to another destination may result in a wrong hop count if the real source takes a different path than replayed traffic. Therefore, HCF is also vulnerable to the replay-fix attack.

HCF's parameter table values change when end-to-end routes change. This occurs very rarely according to [5] and requires a dynamic table update. Quickly and securely detecting legitimate TTL changes can be done for symmetric routes by mining correct parameter values from established TCP connections. *On asymmetric routes, however, it is difficult to recognize established from spoofed TCP connections, thus update of HCF's, RBF's, IDPF's and PiIP's parameter tables upon a routing change is an open problem.* A TTL-guessing attacker creates lower false negatives at a single filter than an attacker that just uses bots' own TTL values [5]. Thus HCF's false negatives from guessing at multiple filters should be lower than $100\% - max(RAP, TP)$ for a given number of filters, i.e. $< 6\%$. Column 2 in Table IV summarizes measures for HCF.

C. Route-Based Filtering

RBF associates each source with the previous hop its traffic crosses to reach the filter. It was proposed as an altruistic defense and its authors recommended a vertex cover deployment [6]. Normalized strength of an RBF filter is: $strength_F = \sum_{p \in IP_{rout}} PH(p)/|IP_{rout}|^2$, where $PH(p)$ is the number of sources whose previous hop at F differs from p's previous hop. ASes with more neighbors should have a higher filtering strength because they have a higher diversity of potential previous hop values.

We show the RBF performance in selfish deployment in Fig. 3(a). Unlike HCF, only a small number of filters have a high *TP* measure in isolated deployment. This is because the AS connectivity follows a power-law distribution so a few ASes are well-connected and make strong RBF filters. As expected,

RAP measure in isolated deployment is low because of low path coverage.

TP and *RAP* measures for altruistic deployment are shown in Fig. 3(b) and 3(c), respectively. Protection of all nodes is similar to that of HCF, and 50 optimal filters result in 93% *TP* and *RAP* measure. Again, the VC deployment offers a slightly higher protection (2-5%) but requires around 60 times more deployment points. Filters' *TP* measure is lower that the same measure for the HCF defense, but it is still higher than an average node's protection in isolated deployment, creating good deployment incentive. The *RAP* measure is the same for filters and for all nodes. Around 22% of IPs are very impaired with regard to attacker placement, and 52% are moderately impaired. RBF has the highest impact on limiting possible attacker locations out of the defenses we evaluated.

RBF has the same per-packet cost and a slightly larger storage cost, when compared to HCF. The storage cost is larger because an AS may have up to several thousand links so two bytes are needed instead of one to store the parameter value. An attacker that shares the path between the source and the filter can spoof this source's traffic to all destinations the source reaches via this path. He can also replay the traffic he captures to the same destinations but this attack is unlikely since spoofing is easier for attackers than replay. RBF is thus vulnerable to the path-all, replay-all and replay-fix attacks.

RBF's parameter table values change when a change in end-to-end routing leads to a previous hop change for some sources. Thus frequency of false positives at an RBF filter is at most as high as for HCF filters, but likely smaller since many end-to-end routing changes may not result in peering change at large ASes that act as RBF filters. An attacker cannot influence the previous hop of spoofed packets with regard to filter thus false negatives due to guessing are zero. Column 3 in Table IV summarizes measures for RBF.

D. Inter-domain packet filtering

IDPF associates each source with a set of feasible neighbors (previous hops). A neighbor N is feasible for source x if N advertises a route to x to this filter. In [7] authors assume that route advertising rules are based on relationships between ASes [15]. IDPF was proposed as an altruistic defense with a recommended vertex cover deployment [7].

Normalized strength of the IDPF filter is: $strength_F = \sum_{p \in IP_{rout}} NF(p)/|IP_{rout}|^2$, where $NF(p)$ is the number of source IPs whose previous hop does not exist in the feasible neighbor set of p. Well-connected nodes are good candidates for strong filters because of the diversity of their neighbors and the prevalence of peer relationships that limit the size of the feasible neighbor set. Like in the case of RBF, because AS degrees follow the power-law distribution we expect the number of strong IDPF filters to be low. This is confirmed by the IDPF performance in selfish deployment in Fig. 4(a). The protection of filters is very low both for the *TP* and for the *RAP* measure.

In optimal deployment, 50 filters provide 80% *TP* measure for all nodes (Fig. 4(b)) and 72% *RAP* measure (Fig. 4(c)). The VC deployment improves *TP* measure to levels comparable with RBF. Due to large memory requirements we could not compute the *RAP* measure for VC. IDPF makes 20% of attacker locations very impaired, and 35% moderately impaired.

The IDPF per-packet cost consists of the lookup of the packet's previous hop in the feasible neighbor set (< 8 ns). The storage cost is about six times higher than RBF's and HCF's because multiple feasible neighbors must be recorded for each parameter table entry. If the average number of neighbors is 280, and each neighbor's feasibility is indicated by a single bit, we need 35 bytes for parameter values plus 5 bytes to store a prefix for each table entry. IDPF is vulnerable to the same attacks as RBF: path-all, subnet-all, replay-all and replay-fix. Additionally, IDPF is also vulnerable to possible-path-all attack, since it will fail to filter packets from attackers that lie on a different path than the source but arrive from a feasible neighbor.

IDPF's parameter table values change when a change in end-to-end routing leads to a change in feasible neighbor sets. The frequency of false positives at an IDPF filter should be higher than at an RBF filter but smaller than at an HCF filter. An attacker cannot influence the feasible neighbor sets of spoofed packets thus false negatives due to guessing are zero. Column 4 in Table IV summarizes measures for IDPF.

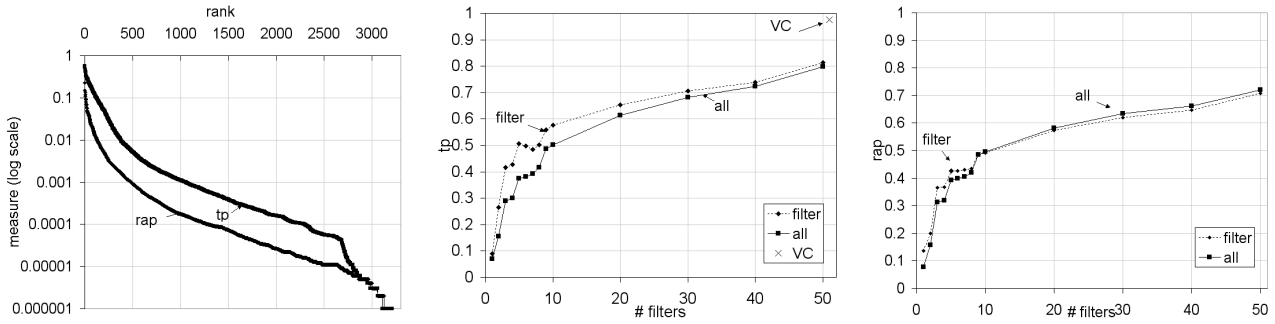
E. Spoofing Prevention Method and Packet Passports

Both SPM and PASS sources place a mark in their outgoing traffic to signify that it is not spoofed. SPM is a selfish but collaborative defense, designed for end-network deployment. An SPM participant filters spoofed traffic only if it is its final destination. The SPM mark is unique to the AS pair and helps prevent spoofing only between them. It is placed into packets in clear.

If SPM participants jointly own $|IP_{spm}|$ addresses, the normalized SPM filter strength is: $strength_F \approx |IP_{spm}|/|IP_{rout}|$. Because the AS size follows the power-law distribution, random selection of SPM participants is unlikely to result in large $|IP_{spm}|$ values, thus sparse, random deployment is unlikely to provide good protection to participants. This is illustrated in Fig. 5(a), which shows the *TP* and *RAP* measures for randomly selected 10, 100 and 1,000 participants. The y-axis shows the average of 10 simulation runs for each random selection, and the error bars show two standard deviations from the mean. The protection is very low and around 5% for both measures and the 1,000 random filters. The reason for such low performance lies in the uneven distribution of the address space.

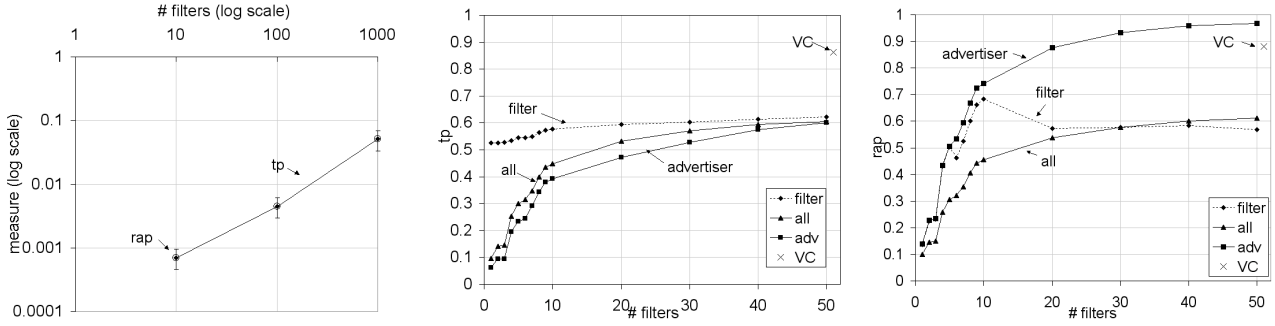
SPM can become an altruistic defense by separating the marking and the filtering functionality. Interested ASes become *SPM-advertisers* while *SPM-filters* are deployed strategically at optimal points. Each advertiser chooses one secret to mark packets (regardless of their destination) and communicates it to all SPM-filters, who use it to filter spoofed traffic. Per-source AS secret instead of per-AS-pair secret is necessary to ensure table scalability. Selfish and altruistic SPM are complementary and can both be deployed to further increase spoofing protection.

To evaluate SPM in the most advantageous scenario we choose 100 largest ASes as SPM-advertisers. The *TP* measures



(a) Isolated deployment: *TP* and *RAP* (b) Optimal deployment: *TP* (c) Optimal deployment: *RAP*

Fig. 4. Performance measures for IDPF



(a) Sparse deployment: *TP* and *RAP* (b) Optimal deployment: *TP* (c) Optimal deployment: *RAP*

Fig. 5. Performance measures for SPM

for filters, advertisers and all hosts are shown in Fig. 5(b). Aggregated AS size of 100 chosen advertisers is around 50%. *TP* for advertisers and for all nodes surpasses this value because of ING deployment at filters. *TP* of filters starts at 50% since spoofing of advertisers' addresses can be removed from all traffic going to a filter, and it increases up to 60% with 50 filters. *RAP* increases for advertisers with the increase in number of filters, because the joint filter popularity increases, and reaches 97% for 50 filters. *RAP* for filters fluctuates as more filters are added, because some early filters are also advertisers. *RAP* for all nodes changes similarly to *TP* measure and reaches 61% with 50 filters. VC deployment of filters offers around 87% protection in both dimensions. Only about 5% of attacker locations are very impaired and 15% are moderately impaired with SPM.

A PASS participant filters all spoofed packets it can identify, both in incoming and in transit traffic. Its effectiveness is thus equivalent to altruistic SPM.

The cost of altruistic SPM includes per-packet lookup ($<$ 8 ns) and 300 KB for the storage cost, assuming that we need two keys per entry (one old, one new, to accommodate late packets), each key is 4 B long and we need 2 B to specify the AS number. Altruistic SPM does not bind the mark to the packet or the destination, so it is vulnerable to the sniff-all and replay-all attacks. Selfish SPM has one mark per destination, so it is vulnerable to the sniff-fix and replay-fix attacks.

A PASS mark is unique to a source-transit AS pair, and is cryptographically bound to the packet, which changes its cost and security properties when compared to SPM. According to [9], the marking cost per advertiser is around 1 μs per-packet and the verifying cost per filter is 250 ns per-packet. The storage cost is significant requiring 60 B of additional packet space and 33 MB at the filter for Bloom filter and key storage. The expressions for the packet and storage cost are taken from [9]. While most expensive, PASS defense offers highest security and is not vulnerable to any of our surveyed attacks.

SPM's parameter table changes upon explicit notification from advertisers, thus false positives should be zero. Since SPM keys are 32-bits long and changed frequently an attacker has negligible chance of guessing a key correctly, and false negatives due to guessing are close to zero. PASS's false positives should be zero due to parameter change, but use of Bloom filters adds an additional source of false positives. In [9] false positive rate was estimated as $5.7*10^{-6}$. Similarly to SPM, PASS's false negatives due to guessing should be zero. Columns 5 and 6 in Table IV summarize measures for SPM and PASS.

F. PiIP

PiIP associates sources with marks denoting routers (markers) between the source and the filter. Normalized strength of a PiIP filter is: $strength_F = \sum_{p \in IP_{rout}} Mark(p)/|IP_{rout}|^2$, where $Mark(p)$ is the number sources whose mark at the filter differs from p's mark.

PiIP was proposed as a selfish defense, and the necessity of placing markers at remote routers makes it a collaborative defense. We evaluated PiIP in sparse, random deployment

using the same approach as in SPM's evaluation. For space reasons we omit this graph, but it looks similar to the SPM's sparse, random deployment graph. *TP* and *RAP* are very small because random marker selection has low chance of choosing popular ASes, thus spoofed traffic is mostly unmarked and cannot benefit from the defense.

PiIP needs no modification to become an altruistic defense. Optimal marker placement should maximize mark distinctiveness for all Internet nodes, while placing markers on popular paths to maximize the number of marked packets. We select optimal markers using the following heuristic: (1) Sort ASes by their popularity pop. (2) Choose the AS X with the largest pop to be a marker. (3) For each neighbor N of a marker X choose N to be a marker if $pop(N) > 0.5 \cdot pop(X)$. If N is chosen, recursively repeat step (3). Repeat steps (2) and (3) until all markers are placed. This heuristic starts with popular ASes as markers and grows this region toward edge ASes to increase distinctiveness of marks. We place total of 100 optimal markers and observe filters as separate from markers.

Again, for space reasons, we do not show graphs for *TP* and *RAP* measures for PiIP in the optimal deployment with 100 optimal markers and 1–50 optimal filters. These measures are almost identical to HCF's performance. PiIP makes 2% of attack locations very impaired and 61% moderately impaired.

The marker's cost is < 8 ns, for placing a deterministic mark into the packet header. The filter's cost consists of a per-packet lookup (< 8 ns) and 60 KB of storage, assuming that we need 2 bytes per entry to store the AS number and additional 2 bytes for the mark. Since the marks are stacked in the IP ID field no additional packet space is needed. Attackers that share the path with the source can spoof this source to all destinations. This includes subnet-all and path-all attacks. Attackers can also replay traffic to any destination, so PiIP is vulnerable to replay-all and replay-fix attacks.

Any change in end-to-end routing is likely to change some fields in PiIP's parameter table, thus its false positives should be comparable to HCF's. Because PiIP's marks in packets are shifted during forwarding any forgery by an attacker is likely to be lost in transit, making false negatives from guessing close to zero. Column 7 in Table IV summarizes measures for PiIP.

G. Comparative analysis

We now summarize the effectiveness results. The first 6 rows in Table IV show the benefit to participants in isolated, single-filter deployment for those defenses that can support isolated deployment. HCF is the only defense that successfully protects filters from spoofed traffic in isolated deployment – *TP* measure is high for all filters, ranging from 94% to 98%. Other defenses show high discrepancy between the best, the median and the worst *TP* measure. This is because the *TP* measure in isolated deployment depends only on filter strength, and except for HCF, all other defenses have very few strong filters due to dependence of their strength on topology features that follow power-law distribution. Note that isolated *RAP* measures are the same for all defenses, i.e., the highest value is 22% and the median and the lowest values are very low.

The next two rows in Table IV show the benefit to participants in random filter deployment on 100 nodes for collaborative defenses only. This benefit is very low, because the filter strength of SPM, PASS and PiIP depends on topology features that follow power-law distribution and random filter selection cannot result in strong filters.

The next two rows in Table IV show the *TP* and *RAP* measures for all nodes for 50 optimally placed filters. In the case of PiIP and SPM/PASS, 100 optimal markers, and 100 optimal advertisers, respectively, were also deployed. Altruistic versions of HCF, RBF and PiIP have a comparable performance across both measures and offer significant protection to all nodes against spoofed and reflected traffic with 50 optimal filters. IDPF, SPM and PASS have lower protection (60-80%), and ING filters eliminate only 30% of spoofed and reflected traffic. Next two rows show the percentage of attacker locations (IPs) that are very impaired and moderately impaired, for 50 optimally placed filters. RBF and IDPF have the highest percentage of very impaired locations – more than 20%. HCF and RBF have the highest impact overall – making around 75% of locations either very or moderately impaired, followed by PiIP and IDPF. ING, SPM and PASS have the lowest impact on attack locations (20-30%).

The rest of Table IV show defense cost, security, false positives and false negatives. Per-packet processing and storage costs are acceptable for all defenses. PASS is the most secure defense, followed by HCF where probing and replay can bypass filters only to a given destination. RBF, IDPF, SPM and PiIP are susceptible to attacks from carefully positioned attackers, via sniffing and replay, to all destinations, while ING only marginally reduces an outside attacker's spoofing choices. False positives due to routing changes are a concern for HCF, RBF, IDPF and PiIP, and approaches to quickly update these defenses' parameter tables upon a route change are an open research problem today. False positives due to other issues exist only in case of PASS and are very low. False negatives due to guessing are close to zero for all defenses, except for HCF, where they are less than 6%.

VI. Impact of Routing and Topology on Performance

Our evaluation showed that defense performance is strongly linked to the inherent properties of the Internet topology – the power-law distributions of AS connectivity, size and popularity. Because these properties are preserved in Internet topologies inferred from various sources [27], we expect that our results will not depend on the chosen source.

We emphasize that all our evaluation and conclusions relate to *AS-level* topologies. Router-level topologies within an AS do not follow power-law as shown by [28] and issues relating to deployment of defenses within an AS require further research. We also emphasize that when we talk about the "power-law nature of topology" we simply mean "a few ASes have large size/large popularity/high connectivity" and not that these features strictly follow the power-law distribution.

A. Topology Impact

We investigated a hypothesis that the power-law nature of a topology is the only deciding factor for spoofing defense

TABLE IV
SUMMARY OF RESULTS FOR SPOOFING DEFENSE PERFORMANCE.

Category	ING	HCF	RBF	IDPF	SPM	PASS	PiIP
Effectiveness in isolated, single-filter deployment showing benefit **to participants**							
highest *TP*	6%	**98%**	87%	57%	N/A	N/A	N/A
lowest *TP*	0.0001%	**94%**	0.0001%	0.0001%	N/A	N/A	N/A
median *TP*	0.0005%	**96%**	34%	0.03%	N/A	N/A	N/A
highest *RAP*	22%	22%	22%	22%	N/A	N/A	N/A
lowest *RAP*	0.0001%	0.0001%	0.0001%	0.0001%	N/A	N/A	N/A
median *RAP*	0.006%	0.006%	0.006%	0.006%	N/A	N/A	N/A
Effectiveness in deployment at random 100 nodes, showing benefit **to participants**							
TP	N/A	N/A	N/A	N/A	0.45%	0.45%	3.3%
RAP	N/A	N/A	N/A	N/A	0.3%	0.3%	0.6%
Effectiveness in deployment at 50 optimal nodes, showing benefit **to all Internet nodes**							
TP	30%	**95%**	**93%**	80%	60%	60%	**94%**
RAP	30%	**95%**	**93%**	72%	61%	61%	**92%**
AI (very)	9%	**4%**	**22%**	20%	5%	5%	**2%**
AI (moderately)	20%	**71%**	**52%**	35%	15%	15%	**61%**
Cost (F - filter cost, M - marker cost, P - packet space cost)							
Per-packet cost expression	l	l	l	$l \cdot N_l$	**F:** l **M:** m_d	**F:** $v_c + l$ **M:** m_c	**F:** l **M:** m_d
Per-packet cost value	8 ns	8 ns	8 ns	2.2 μs	**F:** 8 ns **M:** 8 ns	**F:** 258 ns **M:** 1 μs	**F:** 8 ns **M:** 8 ns
Storage cost expression	$N_p \cdot 5$B	$N_T \cdot 6$B	$N_T \cdot 7$B	$N_T \cdot (5 + N_l/8)$B	**F:** $(N_{AS} \cdot 10)$B	**P:** $(N_{ASh} \cdot 10 + 20)$B **F:** $N_{AS} \cdot 67.5\text{KB} + 32\text{MB}$	**F:** $N_{AS} \cdot 4$B
Storage cost value	25B	1.2MB	1.4MB	7.8B	**F:** 300KB	**P:** 60B **F:** 33MB	**F:** 120KB
Security for altruistic defenses (vulnerable to following attacks)							
spoof-all	yes						
possible-path-all	yes			yes			
path-all	yes		yes	yes			yes
subnet-all	yes		yes	yes			yes
sniff-all	yes				yes		
replay-all	yes		yes	yes	yes		yes
replay-fix	yes	yes	yes	yes			yes
probe-fix		yes					
False positives per table entry							
On e2e route change	0	$h \approx 0$	r $(r < h)$	i $(r < i < h)$	0	0	p $(p \approx r)$
Other	0	0	0	0	0	$5.7 * 10^{-6}$	0
False negatives from guessing (percentage of randomly spoofed packets with correct parameter value/source IP combination)							
	0	$< 6\%$	0	0	≈ 0	≈ 0	≈ 0

performance by repeating our evaluations on a collection of different topologies: (1) **UCR-Extra**: UCR topology ($\approx$ 20 K nodes, 120 K links, 40 K P2P links), with 120 K P2P links added at random, but avoiding highly-connected nodes, (2) **UCR-Uni**: UCR topology with uniformly distributed AS sizes, (3) **WHOIS**: topology inferred from [19], (4) **Skitter**: topology inferred from [20], (5) Inet-generated power-law topology with 20 K nodes and (a) uniform AS sizes (**Inet-Uni**), and (b) AS sizes inferred from the UCR topology (**Inet-UCR**), (6) Randomly generated topology with 20 K nodes and (a) uniform AS sizes (**Rand-Uni**), and (b) AS sizes inferred from the UCR topology (**Rand-UCR**). Inet and random topologies were generated so that around 2/3 of nodes are edges, just like in the UCR, Skitter and WHOIS topologies. Routing in Inet and random topologies is shortest-path. When generating a random topology non-edge nodes were randomly connected until the average number of links matched the value in UCR topology – 6 links. When assigning AS sizes inferred from UCR topology to Inet or random topologies, we ordered nodes by connectivity and ordered AS sizes, then performed assignment by rank so that well-connected nodes have larger AS size; this mimics the correlation observed in the UCR, Skitter and WHOIS topologies.

Table V summarizes our results for *TP*, *RAP* and *AI* (very + moderately impaired) measures for 50 optimally placed filters. Entries where performance declines drastically are marked with stars. We lacked AS relationship information to calculate IDPF's performance, and we omitted PASS since it performs exactly like SPM.

All defenses perform comparatively well in topologies where both connectivity and AS size follow power law (UCR, UCR-Extra, Skitter, WHOIS, Inet-UCR). Topologies where connectivity follows power law but AS size is uniformly distributed (UCR-Uni, Inet-Uni), drastically reduce effectiveness of ING and SPM/PASS because their filter strength depends directly on AS size distribution. Rand-UCR topology, where connectivity does not follow power law but AS size does, decreases effectiveness of RBF, HCF and PiIP because it lacks highly-connected, popular nodes and Gaussian hop-count distribution. Random topology with uniform AS size distribution makes all defenses perform very poorly because neither connectivity nor AS size follow power law.

Quintupling the number of P2P links in UCR-Extra topology did not significantly change either *TP* or *RAP* measures, but *AI* measures for HCF and PiIP increased. No change in protection measures is due to the fact that addition of this many P2P links does not significantly change routing at a macro level, although it does change it for specific destinations. The rise in *AI* measures is likely due to higher path diversity which increases strength of HCF and PiIP filters.

We now comment on topology features that influence a defense's effectiveness in altruistic deployment, and how they may change in the future. Filter popularity has the highest impact, and it is linked both to topology features (a small number of highly connected nodes) and routing features (current prevalence of P2P links makes most traffic go over provider nodes). Recent research shows that P2P links are becoming more popular among lower-tier providers and stubs [24], which should increase connectivity of these nodes, and drive their traffic away from Tier-1 providers. This would increase the number of defense deployment points needed for the same joint popularity and the same defense effectiveness. From our experiments with UCR-Extra topology it is clear that a huge number of P2P links would need to be added to achieve dramatic effect. This is because a single P2P link only affects traffic between its anchors and their customers, but not traffic anchors' and customers' exchange with the rest of the Internet. Until the majority of destinations, from the majority of sources, can be reached via only P2P and customer links, most routes will go up the provider-customer hierarchy leading to a few highly popular ASes.

The counter effect of adding more links to AS topology is the increase in strength of RBF and IDPF filters – due to increased connectivity, strength of HCF filters – due to balancing/lowering of the Gaussian hop distribution, and strength of PiIP filters – due to increased path diversity. Here the rule of "the more the merrier" applies, i.e., making some filters stronger does not diminish the strength of other filters. There are just more eligible nodes to choose from.

The AS size distribution affects the strength of ING, SPM and PASS filters – topologies where a few ASes own majority of the address space result in a few very strong filters. With the shift to IPv6 and the rise of lower-tier ISPs, the address space distribution could become more balanced, which would diminish filter strength, but we lack means to predict how likely this is to happen.

Summarizing our findings, the trends in Internet topology seem to go towards increasing connectivity, spreading out the traffic and possibly balancing the address space allocation. The first two trends affect all defenses, increasing the number of deployment nodes for a desired effectiveness, but large changes in topology features map to small differences in effectiveness measures. These trends also improve the effectiveness of already strong defenses HCF, RBF, IDPF and PiIP. The third trend may lower the effectiveness of ING, SPM and PASS thus indirectly lowering the effectiveness of other defenses that include ING. This secondary effect seems much smaller than the primary effect, based on our evaluations. For example, effectiveness of ING between UCR and UCR-Uni goes from 35-36% to 1%, but effectiveness of RBF goes from 93% to 84%, and HCF's stays the same. Overall, it seems that the relative relationship between defenses and our general conclusions would still hold in the Internet of the future.

B. Routing Impact

We now explore an alternative routing scenario that may affect defense performance: multipath, hot-potato routing that sends traffic on multiple routes between a source and a filter. We evaluate its impact on two chosen defenses, one route-dependent – HCF and one AS-size-dependent – SPM.

TABLE V
SUMMARY OF RESULTS FOR DIFFERENT TOPOLOGIES.

Topology	Measure	ING	HCF	RBF	SPM	PiIP
UCR	*TP*	35%	95%	93%	60%	94%
UCR-Extra	*TP*	34%	95%	93%	60%	94%
Skitter	*TP*	44%	94%	97%	67%	91%
WHOIS	*TP*	55%	92%	93%	73%	90%
Inet-UCR	*TP*	50%	92%	93%	72%	80%
Rand-UCR	*TP*	48%	68% (*)	48% (*)	56%	58% (*)
UCR-Uni	*TP*	1% (*)	95%	84%	2% (*)	91%
Inet-Uni	*TP*	0% (*)	93%	92%	1% (*)	80%
Rand-Uni	*TP*	0% (*)	6% (*)	2% (*)	0% (*)	1% (*)
UCR	*RAP*	36%	95%	93%	61%	90%
UCR-Extra	*RAP*	36%	94%	90%	62%	93%
Skitter	*RAP*	44%	95%	97%	68%	92%
WHOIS	*RAP*	58%	95%	96%	76%	93%
Inet-UCR	*RAP*	52%	94%	94%	74%	99%
Rand-UCR	*RAP*	49%	69% (*)	62% (*)	58%	41% (*)
UCR-Uni	*RAP*	1% (*)	95%	84%	3% (*)	95%
Inet-Uni	*RAP*	0% (*)	93%	92%	1% (*)	98%
Rand-Uni	*RAP*	0% (*)	6% (*)	4% (*)	1% (*)	2% (*)
UCR	*AI (v+m)*	29%	75%	74%	20%	63%
UCR-Extra	*AI (v+m)*	28%	86%	74%	20%	74%
Skitter	*AI (v+m)*	44%	87%	99%	41%	64%
WHOIS	*AI (v+m)*	41%	70%	69%	38%	50%
Inet-UCR	*AI (v+m)*	43%	65%	67%	42%	50%
Rand-UCR	*AI (v+m)*	42%	39% (*)	36% (*)	39%	34% (*)
UCR-Uni	*AI (v+m)*	1% (*)	91%	52%	1% (*)	58%
Inet-Uni	*AI (v+m)*	0% (*)	69%	65%	0% (*)	53%
Rand-Uni	*AI (v+m)*	0% (*)	0% (*)	0% (*)	0% (*)	0% (*)

When deploying SPM, we select 5 optimal advertisers, which now contain around 48% of the address space. With 50 optimal filters the *TP* measure is very high both for filters (87%) and for advertisers (80%), and much larger than corresponding measures on the UCR topology. We attribute this to a smaller total address space in the MP topology. There are about 10,000 ASes in the MP topology that contain around 300 million addresses. This is $1/2$ of the AS count and $1/5$ of the addresses from the UCR topology. Thus the effect of ING deployment at 50 popular nodes almost doubles the SPM's effectiveness. We verified this by running SPM without ING, and indeed the *TP* measure was at most 48%. The *RAP* measure is around 90% for advertisers and 61% for filters.

The HCF's *TP* measure for all nodes is lower than for the UCR topology, due to multiple parameter values being associated with a source address, but is still high – around 85% for 50 filters, as compared to 95% in the UCR topology. The filter's *TP* measure is around 90%. The *RAP* measure is similar to the one for the UCR topology – filters and all nodes receive the same protection – 93% with the 50 optimal filters. We also evaluated HCF without ING, and its performance was about 10% lower in the MP topology and 5% lower in the UCR topology. Thus, continued good performance of HCF does not stem from a bias present in the small MP topology toward large filters, but from inherent topological properties (a few very popular paths, Gaussian hop count distribution) that exist even with multipath forwarding.

C. Deployment Strategy Impact

For realistic deployment of altruistic defenses, calculation of optimal deployment points would be a problem because it

requires full knowledge of global routing patterns. However, there is a strong correlation between a node's popularity, connectivity and AS size, so one of connectivity/size criteria could be used selection of deployment points. We investigated performance of altruistic defenses deployed at top 1–50 ASes in terms of (1) AS size and (2) connectivity. While alternative deployment strategies performed worse than optimal at 1–20 filters, the performance was the same as deployment increased above 20 filters.

Finally, we consider a realistic scenario in which a modest subset of tier-1 ISPs deploy the HCF or the SPM defense. Querying the ARIN database [29], we found that 18 organizations contribute majority of filters in the optimal set for any defense: AOL, AT&T, APNIC, Bell Canada, Beyond The Network America, California State University Network, Cogent Communications, DoD, Global Crossing, Level 3, MCI, NTT America, Qwest, RIPE, Savvis, Sprint, Teleglobe and Time Warner. We then simulated deployment of HCF or SPM on all ASes that belong to these 18 organizations within the UCR topology (221 ASes total). HCF protected deploying organizations from 97% of spoofed traffic and 97% of reflected traffic. All nodes received 96% protection from spoofed and reflected traffic. When deploying SPM, the chosen organizations were both the advertisers and the filters. SPM protected participants from 53% of spoofed and 98% of reflected traffic. Both measures were at 53% protection for all Internet hosts. Thus HCF or SPM would significantly decrease spoofing in the Internet if deployed only by the 18 tier-1 organizations. We expect that the performance of other altruistic defenses would be similar in this deployment scenario.

VII. Conclusions

Our evaluation shows that edge network defenses, deployed either in isolated or collaborative manner, cannot offer sufficient protection without core support. Only HCF offered significant protection in selfish, isolated deployment, but only against spoofed traffic and not against reflector attacks. Collaborative defenses – SPM, PASS and PiIP – had very poor performance when deployed sparsely at random, offering low motivation for early adopters. Such performance is not the consequence of current defense design but of edge deployment. Any novel edge network defenses are likely to fail in the same manner.

Results speak strongly in favor of a systematic, Internet-wide deployment of spoofing defenses at strategically positioned ASes. SPM, PiIP and HCF were much more effective as altruistic than as selfish defenses, and only altruistic defenses achieved sufficient path coverage to reduce reflector attacks. Prior research [6], [7] proposed vertex cover filter deployment for Internet-wide protection. Our results show that a much lower deployment at 50 optimally selected ASes can achieve a comparable, and sometimes better effectiveness. Novel research should focus on improving cost and security of effective defenses such as HCF, RBF and PiIP, designing algorithms for parameter table update, and investigating defense combinations and novel altruistic defenses.

References

[1] Advanced Network Architecture Group. ANA Spoofer Project. http://spoofer.csail.mit.edu/.

[2] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. IETF RFC 2267.

[3] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage. Inferring Internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2), May 2006.

[4] D. Kawamoto. DNS recursion leads to nastier DoS attacks. ZDNet.co.uk, 17 March 2006.

[5] C. Jin, H. Wang, and K.G. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proc. of the 10th ACM conference on Computer and communications security*, 2003.

[6] K. Park and H.Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets. In *Proc. of ACM SIGCOMM*, 2001.

[7] Z. Duan, X. Yuan, and J. Chandrashekar. Constructing Inter-Domain Packet Filters to Control IP Spoofing Based on BGP Updates. In *Proc. of INFOCOM*, 2006.

[8] A. Bremler-Barr and H. Levy. Spoofing Prevention Method. In *Proc. of INFOCOM*, 2005.

[9] X. Liu, X. Yang, D. Wetherall, and T. Anderson. Efficient and Secure Source Authentication with Packet Passports. In *Proc. of SRUTI*, 2006.

[10] A. Perrig, D. Song, and A. Yaar. StackPi: A New Defense Mechanism against IP Spoofing and DDoS Attacks. Technical Report CMU-CS-02-208, CMU Technical Report, February 2003.

[11] M. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, and M. De Shon. Predicting future botnet addresses with uncleanliness. In *Proc. of IMC*, 2007.

[12] V. Yegneswaran, P. Barford, and S. Jha. Global Intrusion Detection in the DOMINO Overlay System. In *Proc. of NDSS*, 2004.

[13] Y. He, G. Siganos, M. Faloutsos, and S. V. Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and Impact. In *Proc. NSDI 2007*, April 2007.

[14] RouteViews.org. BGP Core Routing Table Size. http://www.routeviews.org/dynamics/.

[15] F. Wang and L. Gao. On inferring and characterizing Internet routing policies. In *Proc. Internet Measurement Conference*, October 2003.

[16] University of Oregon. RouteViews Archive. http://www.routeviews.org.

[17] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet Quarantine: Requirements for Containing Self-Propagating Code. In *INFOCOM*, 2003.

[18] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-Topology Model that Captures Route Diversity. In *Proc. of ACM SIGCOMM*, 2006.

[19] Internet Routing Registries. http://www.irr.net.

[20] CAIDA. Skitter data.

[21] The DIMES Project. DIMES web page. http://www.netdimes.org/.

[22] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *ACM SIGCOMM CCR*, January 2005.

[23] G. Siganos and M. Faloutsos. Analyzing BGP Policies: Methodology and Tool. In *INFOCOM*, 2004.

[24] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In Search of the Elusive Ground Truth: the Internet's AS-level Connectivity Structure. In *Proceedings of ACM SIGMETRICS*, 2008.

[25] D. S. Hochbaum. Approximation Algorithms for NP-Hard Problems. 1996.

[26] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of Networks: From Biological Nets to the Internet and WWW. *Oxford University Press*, 2003.

[27] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K. Claffy, and A. Vahdat. The Internet AS-level Topology: Three Data Sources and One Definitive Metric. Technical report, UCSD, 2005.

[28] D. Alderson, L. Li, W. Willinger, and J. C. Doyle. Understanding Internet Topology: Principles, Models, and Validation. *IEEE/ACM Transactions on Networking*, 13(6):1205–1218, 2005.

[29] American Registry for Internet Numbers. WHOIS. http://www.arin.net/whois.

Dr. Jelena Mirkovic is a Computer Scientist at the USC Information Sciences Institute, which she joined in 2007. Prior to this she was an Assistant Professor at the Computer and Information Sciences Department, University of Delaware, 2003-2007. She received her M.S. and Ph.D. from UCLA, and her B.S. in Computer Science and Engineering from the School of Electrical Engineering, University of Belgrade, Serbia. Her current research is focused on scientific cyber security experimentation, safe sharing of network data, denial-of-service attacks and IP spoofing. Her research is funded by the National Science Foundation, the Department of Homeland Security and the Infosys Corporation.

Ezra Kissel received the BSc degree in Computer Science from the University of Delaware in 2003. After working in private industry for close to 2 years, he returned to the University of Delaware earning the MSc degree in 2007 and is currently a PhD candidate in the CIS department. His research interests include high-performance networking, network protocol design, grid computing, and network security. He has served as a volunteer for SCinet at Supercomputing 08/09 and has participated in the Bandwidth Challenge.