Comparative Evaluation of Spoofing Defenses

Ezra Kissel, University of Delaware and
Jelena Mirkovic, USC/ISI

ABSTRACT

IP spoofing exacerbates many security threats. While many contemporary attacks do not exploit spoofing, a large number still do — thus eliminating or reducing spoofing would greatly enhance Internet security. Seven spoofing defenses have been proposed to date; three defenses 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 deployment points to achieve best protection at lowest cost.

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 specifying different performance goals for each type, (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 valuable insights into advantages and limitations of the proposed defenses, and uncover the relationship between any spoofing defense's performance and the Internet topology features.

1. INTRODUCTION

IP spoofing is a frequent tool in distributed denial-of-service (DDoS) attacks and intrusions. It is also necessary for *reflector* DDoS attacks, where an attacker spoofs a victim's address in service requests sent to a large number of legitimate servers, called "reflectors," that then flood the victim with replies. IP spoofing exacerbates all these threats: because source addresses cannot be trusted, a simple defense approach that remembers and penalizes known offenders cannot be applied.

Many researchers believe that spoofing is a solved problem based on two arguments: (1) a recent Spoofer project's study [1] that estimates that 80% of networks deploy ingress filtering and (2) prevalence of non-spoofed DDoS attacks.

The Spoofer project [1] uses a set of distributed agents that spoof packets to a monitoring machine to measure the prevalence of ingress filtering in networks hosting the agents. Ingress filtering [2] removes outgoing traffic that spoofs addresses outside the deploying network's address range. Spoofer measurements show that around 80% of networks participating in the project deploy ingress filtering. Because the total number of probed addresses is about 15% of assigned IP addresses today, these results cannot be readily extrapolated to the entire Internet. We further emphasize that even if only 20% of all networks in the Internet allowed spoofing, they could still generate high-volume spoofed and reflected traffic to any target; a fact also noted by the authors of the Spoofer project in [3]. Spoofing is thus very much an open problem in today's Internet.

Contemporary DDoS techniques have gravitated toward flash-crowd type attacks that do not use spoofing. Still, there are a significant number of DDoS attacks that take advantage of spoofing. For example, recent analysis of backscatter traffic [4] used replies to spoofed packets to infer that there were several hundred DDoS attacks with IP spoofing per day in 2001—2004, and via personal communication with the study's authors we confirmed that the data for 2005 and 2006 look similar. Thus, while there may be many other DDoS attacks that do not use spoofing and could not be observed in backscatter traffic, it is certain that several hundred attacks per day use spoofing — a large number to be discarded as insignificant. Another type of DDoS attacks that has recently become popular are reflector attacks — these attacks must use spoofing and will not be visible in backscatter traffic because the replies go directly to the victim. In the two-month interval between December 2005 and February 2006 about 1,500 targets were attacked via recursive DNS attacks that deploy spoofing for reflection and amplification [5]. Such attacks are devastating to targets that must allow DNS traffic for their daily activities. Eliminating or reducing spoofed traffic would thus greatly improve the state of Internet security.

To date, seven filtering approaches have been proposed to detect and discard spoofed traffic: ingress filtering (ING) [2], hop-count filtering (HCF) [6], route-based filtering (RBF) [7], inter-domain packet filtering (IDPF) [8], spoofing prevention method (SPM) [9], packet passports (PASS) [10] and StackPI (SPi) [11]. ING, HCF, and SPM were proposed for end network deployment, while RBF, IDPF, PASS, and SPi require collaboration with core networks for filtering or packet marking. Because the proposed approaches differ greatly in their design, deployment patterns and performance goals, they have been evaluated by their authors in a customized setting and using custom performance measures. This hinders comparison of proposed defenses. Specifically, the following questions remain unanswered: (1) What end-network defenses can offer good protection in *isolated* deployment? (2) What collaborative end-network defenses can offer good protection in *sparse, random* deployment? (3) What is the best achievable protection from core defenses in *sparse, strategic* deployment? (4) What is the optimal strategy for core-defense deployment that yields best protection with fewest deployment points? These questions address the defenses' effectiveness. It is further important to understand the costs of each defense, and its ability to function well in presence of attackers familiar with its design.

This paper tackles the problem of spoofing defense evaluation in a common setting. We first formalize the analysis of defense effectiveness and define comprehensive performance measures. We then apply this common evaluation framework to the seven defenses proposed to date, to comparatively evaluate their performance. We specifically focus on identifying *practical* defenses — those that can either offer significant protection to the deploying network in isolated deployment, or that can offer significant protection to a large number of Internet hosts in very sparse core deployment (up to 50 core participants).

1.1 Contributions

1. We define a theoretical framework for performance analysis of spoofing defenses (Section 2). Each packet is observed as a tridimensional variable, with the dimensions: real source s, destination d, and spoofed source address p. The main goal of spoofing defenses is to limit possible $\{s,\ d,\ p\}$ combinations that reach the destination. We show that any defense's performance towards this goal 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. For three defenses (SPi, RBF, IDPF) filter strength also depends on the Internet topology — better connected filters have higher impact. For ING, strength depends on the size of the deploying network's address space, while for SPM and PASS it depends on the size of the cumulative address space of all deploying networks. For HCF, filter strength is almost constant and high for any filter. These observations guide our specification of an optimal filter deployment strategy for each defense.

2. We define two intuitive defense performance measures (Section 3) that express the reduction of: (1) spoofed traffic reaching its destination and (2) addresses that can be used in reflector attacks.

3. We evaluate the effectiveness of all filtering defenses proposed to date in a common setting (Sections 4 and 5), replicating Internet topology and routing at the autonomous system level. Our results indicate that three defenses (HCF, RBF, and SPi) would bring significant spoofing reduction to all Internet users, and across all dimensions, if deployed at 50 strategically selected autonomous systems. Spontaneous defense deployment, however, can only protect deploying networks against spoofed traffic, but not against reflector attacks. This protection is moderate to poor, for all defenses but HCF. Thus, spoofing can only be eliminated if there is a strategic filter deployment at selected networks.

4. Our methods are easily applicable to future defenses.

Since the filter popularity of all defenses, as well as the filter strength of several others, depend on the Internet topology and routing, it appears at first blush that our choice of AS topology and routing will critically influence the conclusions we draw from the evaluation. Since there is no common agreement in the research community about the "right" model of AS topology and routing, we are left with a difficult situation. We discuss several topology and routing inference approaches in Section 4.1 and argue for the approach we adopted in this paper. In Section 6 we discuss the impact a different topology or routing would have on our results. We conclude the paper in Section 7.

2. THEORETICAL ANALYSIS OF FILTER EFFECTIVENESS

Let IP_{allc} and $IPv4$ be the set of allocated 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 $s \in IP_{allc}$ to destination address $d \in IP_{allc}, d \neq s$, 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. Because IP_{allc} and $IPv4$ are very large, we ignore the $d \neq s$ and $p \neq s$ restrictions in the analysis, to obtain less cluttered formulas, but we honor them during the evaluation in Section 5.

2.1 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:

$$hit_F(s,d) = \begin{cases} 1, & \text{path from } s \text{ to } d \text{ contains F,} \\ 0, & \text{otherwise} \end{cases}$$

All filtering approaches detect spoofed packets by building a table that associates source addresses (aggregated at some granularity) with some parameter as summarized in Table 2.1. The 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, e.g., sharing of previous hop parameter at a filter will occur if paths from several sources to this filter overlap. 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:

$$diff_F(s,p) = \begin{cases} 1, & \text{F can prevent } s \text{ from spoofing } p, \\ 0, & \text{otherwise} \end{cases}$$

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
SPi	Sequence of packet marks, each router places one mark.

Table 1: Parameter associated with a source IP

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***:

$$filter_F(s,d,p) = hit_F(s,d) \cdot diff_F(s,p), \tag{1}$$

and we define the ***filter impact*** of F as the number of all possible $\{s,d,p\}$ combinations that are filtered by F:

$$impact_F = \sum_{s \in IP_{allc}} \sum_{d \in IP_{allc}} \sum_{p \in IPv4} filter_F(s,d,p) \tag{2}$$

We now delve deeper into components of filter impact to understand how a filter's performance relates to filter deployment and the characteristics of the chosen defense approach. 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:

$$strength_F(s) = \sum_{p \in IPv4} diff_F(s,p), \tag{3}$$

and we define ***filter strength*** as the aggregate of filter strength per source for all sources:

$$strength_F = \sum_{s \in IP_{allc}} strength_F(s) \tag{4}$$

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:

$$pop_F(s) = \sum_{d \in IP_{allc}} hit_F(s,d), \tag{5}$$

and we define ***filter popularity*** as the aggregate of filter popularity per source for all sources:

$$pop_F = \sum_{s \in IP_{allc}} pop_F(s) \tag{6}$$

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

$$impact_F = \sum_{s \in IP_{allc}} pop_F(s) \cdot strength_F(s). \tag{7}$$

Thus both popularity and strength play an important role in defining a filter's impact. Those two quantities are independent of each other, but they interact in defining the impact at the single source granularity. Filters that are likely to have a high impact need not only be popular and strong, but must be *popular and strong for the significantly overlapping groups of sources.*

2.2 Multiple Filters

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

$$\begin{aligned} filter_{FS}(s,d,p) &= \bigvee_{F \in FS} filter_F(s,d,p) \\ &= \bigvee_{F \in FS} hit_F(s,d) \cdot diff_F(s,p) \\ &= \bigvee_{F \in FIL(s,d)} diff_F(s,p), \end{aligned} \tag{8}$$

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. (8) 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:

$$\begin{aligned} impact_{FS} &= \sum_{s \in IP_{allc}} \sum_{d \in IP_{allc}} \sum_{p \in IPv4} filter_{FS}(s,d,p) \\ &= \sum_{s \in IP_{allc}} \sum_{d \in IP_{allc}} \sum_{p \in IPv4} \bigvee_{F \in FIL(s,d)} diff_F(s,p) \qquad (9) \\ &= \sum_{s \in IP_{allc}} \sum_{d \in IP_{allc}} \left| \bigcup_{F \in FIL(s,d)} \{p \mid diff_F(s,p) = 1\} \right| \qquad (10) \end{aligned}$$

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

$$strength_X(s) = \left| \bigcup_{F \in X} \{p \mid diff_F(s,p) = 1\} \right| \tag{11}$$

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:

$$impact_{FS} = \sum_{s \in IP_{allc}} \sum_{d \in IP_{allc}} strength_{FIL(s,d)}(s) \tag{12}$$

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 filters for given cost N we need to select deployment points that lie on many paths, have the ability to detect and discard many $\{s,p\}$ combinations, and the filtered combinations on the path do not overlap significantly.

3. FILTERING PERFORMANCE MEASURES

The main goal of a spoofing defense is to provide protection against two types of threats: (1) spoofed traffic, especially randomly spoofed as part of DDoS attacks, (2) reflected traffic. We now define two intuitive measures of the filtering performance that express these two protection modes. While we developed other measures such as fairness of protection and reduction in number of usable attack sources, we do not present them here for space reasons.

3.1 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 a node x from spoofed traffic, assuming a random deployment of attacking machines and random spoofing.

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

$TP(x)$ depends on the number of filters that lie on routes 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.

3.2 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 a 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 creation of reflected traffic.

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

$RAP(x)$ depends on the path coverage and the filters' ability to detect spoofing of the address x. Thus, isolated defenses and collaborative defenses that are deployed spontaneously cannot in general provide good protection against reflector attacks, because they do not have sufficient path coverage.

4. DEFENSE EVALUATION METHOD

We evaluate effectiveness of the seven 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 [12]. We then use No-Valley-Customer-Prefer approach to infer routing behavior from AS relationships. During evaluation we calculate parameter tables for each defenses, generate packets that traverse $\{s, d, p\}$ parameter space, and calculate performance measures defined in Section 3. 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 measurement is performed and how we specify performance goals for each defense.

4.1 AS Topology and Routing Inference

Ideally, all our calculations should be done on the router-level topology of the entire Internet. Since such a topology is not available, we resort to measurement on the AS-level topology. The following properties of the Internet topology must be faithfully reproduced to guarantee the correctness of the measured quantities: (1) Inter-AS routing and (2) AS address space size. The AS address space size can be easily inferred from the RouteViews data [13] by assigning the size of each unique prefix to the AS that is the last hop on the route to this prefix. Prefixes with multiple origins are few and can be ignored. On the other hand, inference of inter-AS routing is a little short of guesswork due to a lack of necessary information in the public domain. There is no global database of routing tables for the entire Internet. RouteViews data [13] provides such information for a very small number of ASes (around 50 at the time of this writing). This means that we must somehow infer the inter-AS routing from publicly available data about the Internet.

This is especially difficult since routing decisions are made based on complex and private BGP policies and depend on the relationship between an AS and its neighbors, as well as on the routes' properties. One commonly used approach is to infer AS connectivity, and then apply shortest-path routing on this graph [7, 14]. This, however, is naive, unrealistic, and results in incorrect routes, because Internet routing is not shortest-path. Another approach 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, 16]. This principle says that an AS will prefer routes learned from customers and siblings, versus routes learned from peers, and it will prefer peer routes to provider routes. In reality, this principle is used in conjunction with private BGP policies to make routing decisions, so inferred routes may differ from real ones, which is unfortunate. Still, the relationship-based approach to route inference should produce more accurate routes than shortest-path alone. Finally, in [17] Mulbauher et al use iterative learning and multiple quasi routers per AS to learn routing policies that result in the best fit between real and inferred routes. These policies can later be used for prediction of unobserved routes with great accuracy. This approach produces the most realistic routes so far, but we could not use it exclusively in our work for the following reasons: (1) Results of Mulbauher et al study are not yet publicly available, and (2) The study uses information from numerous sources, many of these privately obtained. Thus we could not reproduce their results by reimplementing the learning approach on public data. We did, however, correspond with authors of [17] who have kindly shared with us the routing model for a 1000-node subset of the AS topology. While the size of this sample topology is too small to use it exclusively for spoofing defense evaluation, we do use it in Section 6 to evaluate the impact of multipath routing on our results. In summary, we use the relationship-based route inference approach since this is more realistic than shortest-path approach and relies only on publicly available data. This implies that we need as realistic as possible AS-connectivity map and AS-relationship information to guarantee accurate routes.

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 information source is most complete. In Table 2 we list all the sources of AS connectivity information known to us, and note the advantages and disadvantages of each source. We also specify if a given source provides information about AS relationships.

Since we need the AS relationship data in addition to connectivity to infer routing and also to populate IDPF's parameter tables, we are limited to sources that can provide such data, namely RouteViews [13] and UCR[12]. We decided to use the UCR source since it provides more detailed and accurate AS connectivity and relationship information.

The UCR's connectivity and relationship 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 ob-

Table 2: AS topology sources

Source	Advantage	Disadvantage	Rels?
RouteViews [13]	Some routing data also available	Many AS links missing	yes
WHOIS [18]	Lot of links available that are not present in other sources	Many links are backup, not used by data traffic	no
Skitter [19]	Contains links actually traversed by data traffic	Misses almost half of ASes due to limited number of measurement points	no
NetDimes [20]	Contains links actually traversed by data traffic	May miss some links or nodes if not traversed by probe traffic	no
IRL UCLA [21]	Many links from various sources	May still miss some links or nodes	no
UCR [12]	Many links from various sources	May still miss some links or nodes	yes
Inet [22]	Topologies with realistic properties	Artificial data	no

solete 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.

4.2 Measurement 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, aggregating where possible to reduce computational load. A packet traverses the path from s to d using routing information, and may be filtered by filters that lie on the route. We collect successful and failed packet transmission 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.

4.3 Related Work

Our evaluation approach is similar to those used in [7] to evaluate the RBF defense and in [8] to evaluate the IDPF defense. We also build the AS-level topology from the public routing data, 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 [7, 8] are: (1) We observe s, d and p dimensions at the ***IP-level***, unlike [7, 8] that observe them at the ***AS-level***. Because the IP size distribution across ASes is far from uniform, AS-level analysis leads to false, lower effectiveness results. (2) We define an optimal filter selection strategy for each defense, and evaluate defenses in multiple sparse deployment scenarios (1–50 filters), while [7, 8] focus on a single 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 [7, 8] (e.g. y% of *ASes* cannot receive any spoofed traffic). To our knowledge this is the first work that evaluates several spoofing defenses in a common setting. (4) The AS topology in [7, 8] was inferred from RouteViews and is thus less complete than the topology we use.

In [9] authors evaluate the SPM defense on an artificial and simplified Internet topology at the ISP level — they assume that the Internet contains 100 ISPs, and that their address space size is distributed according to the Zipf law. Since SPM is edge-based defense no routing data was necessary for evaluation. In [6] authors evaluate the HCF defense in an isolated deployment, so only a single filter's protection from spoofed traffic and operation cost were measured. In [11] authors evaluate the SPi defense, focusing on a single victim's protection from DDoS, achieved by packet marking in a given attack scenario. We evaluate spoofing protection offered by SPi to all or participating hosts. The evaluation in [11] is performed using the Skitter topology [19] and the topology from the Internet Mapping project [24]. Since the Internet Mapping project is no longer maintained, we did not include it in our list of topology sources.

4.4 Assumptions

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, this assumption does not hold for 75% of ASes as shown in [17]. There are two ways in which an AS could forward packets on multiple routes to the same prefix: (a) *Deterministic forwarding* within AS — packets from each source of transit traffic arrive at a single router and use this router's path to a destination prefix. In this case each router can be observed as a quasi-AS with a single router and our evaluation approach is still valid but the AS topology has more nodes. (b) *Hot-potato forwarding* or load balancing within AS — each packet could be routed on any of multiple known routes and hence packets from a given source could reach an AS from multiple routes. Hot-potato routing would potentially adversely affect the effectiveness of some spoofing defenses since it would associate multiple parameter values with a single source.

The information about the AS forwarding policies is not publicly available, so it is difficult to decide which forwarding is more realistic. Moreover, we know of no inference method for multiple routes to a prefix that was proven to generate accurate routes. Our decision was thus to take the best of both worlds: (a) We use deterministic forwarding assumption in our main evaluation, with UCR topology, and assume single router per AS. In Section 6 we show the effect that the multipath forwarding has on our conclusions, using a 1000-node topology with routes obtained from [17]. (2) Any node, filter or not, can detect and filter traffic from non-allocated (bogon) address space, e.g. using [25]. This filtering is not part of the effectiveness calculation. Instead, we limit spoofable addresses to $\{p \mid p \in IP_{allc}\}$.

4.5 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, PASS, and SPi are selfish defenses. With slight modifications that we propose in the following sections they can become altruistic defenses. Ingress filtering, RBF and IDPF are altruistic defenses.

When we evaluate a selfish defense, we will focus on the protection enjoyed by each participant (TP and RAP measure) in isolated deployment. This is a realistic assumption as selfish defenses are deployed spontaneously. If isolated deployment is impossible, e.g., because a defense is collaborative and needs at least two participants, we will assume random and sparse deployment. 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. Since vertex cover deployment was proposed for several altruistic defenses [7, 8] we will also evaluate them in this setting for comparison purposes. The vertex cover size is 3394 nodes in our topology.

4.6 Optimal Deployment Strategies

Given a fixed cost of N deployment points, an optimal deployment strategy is such that 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 [26]. Instead, we use a simple greedy heuristic which starts with an empty set of deployment points — D and 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. In our measurements, top 50 filters covered more than 97% of $\{s, d\}$ paths and for some defenses were sufficient to filter more than 95% of spoofed traffic.

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. The probability of selecting an AS or an address prefix as a sample is 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 most frequent N filters as optimal deployment points. Figure 1 shows the pseudo code of this algorithm.

```
i=0
Repeat T (=10) times
    Select k source ASes
    Select k destination prefixes
    Select k spoofed ASes
    C = {}
    D = {}
    NF = all sample combinations
    while D.size < N
        Find AS A that filters most combinations from NF
            D = D + B
            Add filtered combinations to C
            Remove filtered combinations from NF
    Store D as Di, increment i
For each AS in D1...DT, calculate number of sets that contain it
Order ASes by frequency of appearance and select top N
```

Figure 1: Pseudocode of the selection algorithm for N optimal deployment points

5. EVALUATION RESULTS

5.1 Ingress Filtering

Ingress filtering ensures that outgoing packets of a deploying network carry an “inside” source address and incoming packets carry an “outside” source address. Because the external address space is always much larger than the internal one, ingress filtering prevents random-spoofing by the participating networks and is thus an altruistic defense. Normalized strength of a filter with $|IP_{ing}|$ addresses is:

$$strength_F = 2 \cdot |IP_{ing}| \cdot |IP_{allc} - IP_{ing}|/|IP_{allc}|^2 \qquad (13)$$

Strong filters are therefore those ASes that own a large address space.

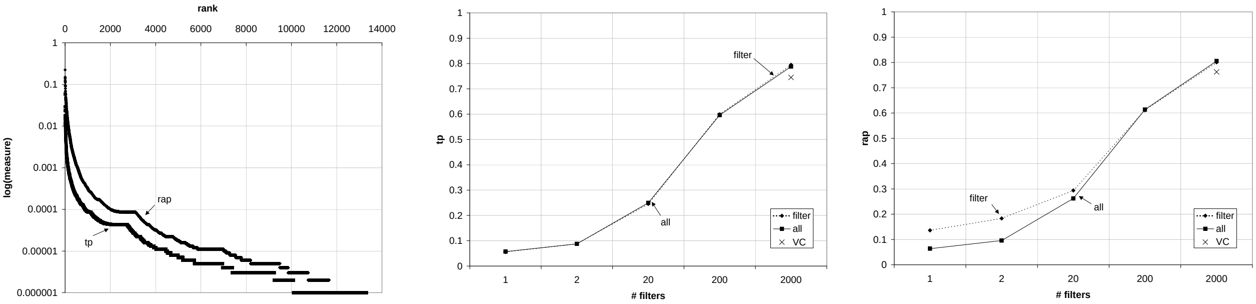
We first evaluate ingress filtering in the isolated deployment for completeness purposes. Figure 2(a) shows TP and RAP measures for an isolated filter on the y axis, on a logarithmic scale. The x axis shows the filter rank. As expected, both TP and RAP are extremely small for majority of filters. The best RAP value is 22% for AS 701, because this AS exists on 22% of source-destination paths and thus cannot be spoofed on these paths. The best TP value is 6%. These results confirm that ingress filtering brings very small benefit to the deploying networks.

For altruistic deployment scenarios, we observe modified ingress filtering that in addition to removing packets with external addresses from outgoing traffic, and packets with internal addresses from incoming traffic, also removes packets with internal addresses from the *transit* traffic. This change increases slightly the benefits of ingress filtering.

Fig. 2(b) and 2(c) show TP and RAP measures for the optimal, random and VC filter deployment, 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. We show the protection of filters and of all Internet hosts separately. Optimal deployment of filters on 2000 nodes (around 6% 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 non-filters 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 non-filters when deployment is sparse (< 200 nodes) since the joint size of filter address space is smaller than the size of the external address space. Each filter's addresses cannot be spoofed by other filters and by the external traffic that hits this filter. On the other hand, addresses of a non-filter cannot be spoofed by filters but can freely be spoofed by anyone else. The amount of external traffic that hits a filter thus creates the difference in the RAP measure. As the number of filters increases this amount becomes smaller and RAP measures become similar.

5.2 Hop-Count Filtering

A hop-count filter associates a source with a router hop count between it and the filter. Hop-counts are inferred from



(a) TP and RAP in isolated deployment.

(b) TP in optimal deployment

(c) RAP in optimal deployment

Figure 2: Performance measures for ingress filtering

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—3] 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, as measured in [6].

HCF was proposed as a selfish defense. Normalized strength of a hop-count filter is:

$$ns_F = \sum_{p \in IP_{allc}} Hop(p)/|IP_{allc}|^2 \tag{14}$$

where $Hop(p)$ is the number of all sources whose hop count differs from p's hop count. In isolated deployment, TP measure is consistently high for all filters, while RAP measure depends on popularity and is low (Fig. 3(a)). HCF can be transformed into an altruistic defense by applying the same filtering approach to the forwarded traffic. *TP* and *RAP* measures shown in Fig. 3(b) and 3(c) are very high for optimal deployment and top 20 filters offer 91% protection.

5.3 Route-based filtering

A route-based filter associates a source address with the previous hop traversed by this source's packets. RBF is altruistic defense and its authors recommended a vertex cover deployment [7]. Normalized strength of an RBF filter is:

$$ns_F = \sum_{p \in IP_{allc}} PH(p)/|IP_{allc}|^2, \tag{15}$$

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. In isolated deployment there are very few nodes with high filter strength (Fig. 4(a)) because AS connectivity follows power law.

Filters' *TP* measure is very high for optimal deployment (Fig. 4(b)) and similar to HCF's. *RAP* measure is same for filters and non-filters (Fig. 4(c)) and depends on the path coverage.

5.4 Inter-domain packet filtering

An inter-domain packet filter associates each source with the set of *feasible* neighbors. A neighbor N is feasible for source x if N advertises a route to x to this filter. In [8] authors assume that route advertising rules are based on relationships between ASes [16]: providers advertise all routes to customers, while customers advertise to providers, and peers advertise to peers, only the routes learned from customers. AS relationships can be inferred from the RouteViews data [15]. IDPF is an altruistic defense and its authors recommended a vertex cover deployment [8]. Normalized filter strength is:

$$ns_F = \sum_{p \in IP_{allc}} NF(p)/|IP_{allc}|^2, \tag{16}$$

where $NF(p)$ is the number of source IPs whose previous hop does not exist in the feasible neighbor set of p.

In isolated deployment, TP measure declines more rapidly than for RBF (Fig. 5(a)), because large feasible sets reduce filtering strength. The RAP measure resembles the RBF's one, and depends mostly on the path coverage.

In optimal deployment, 50 filters offer somewhat lower but still comparable *TP* to RBF (Fig. 5(b)), while the *RAP* measure is barely 60% (Fig. 5(c)).

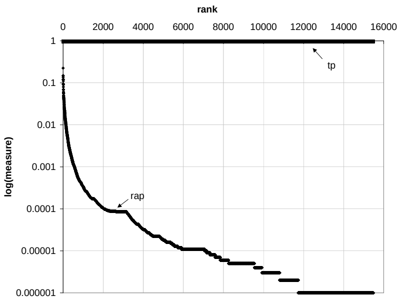
5.5 Spoofing Prevention Method

AS pairs deploying SPM exchange a secret, use it to mark packets they send to each other and filter packets with incorrect marks. The mark is unique to a host pair and helps prevent spoofing only between them. If A and B are the only two networks deploying SPM, A cannot spoof any address to B and vice versa. Additionally, no one can spoof A's addresses to B or B's addresses to A, but they can be freely spoofed to non-SPM participants and the addresses of non-SPM networks can be spoofed to A and B. SPM is a selfish defense, designed for end-network deployment. If SPM participants own $|IP_{spm}|$ addresses, normalized filter strength is:

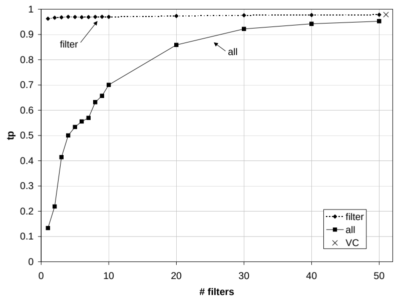
$$ns_F = \sum_{p \in IP_{spm}} |IP_{allc} - IP_{AS(p)}|/|IP_{allc}|^2 \approx |IP_{spm}|/|IP_{allc}|. \tag{17}$$

Because of uneven distribution of IP size, spontaneous SPM deployment would result in small $|IP_{spm}|$ and poor participant protection.

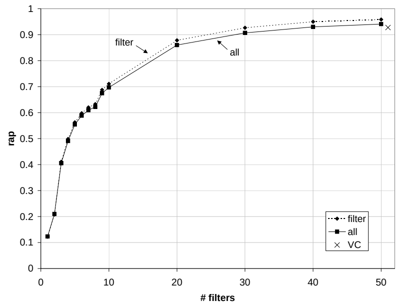
SPM can become an altruistic defense by: (1) Selecting large ASes to be *SPM-advertisers* and deploying *SPM-filters* on popular nodes. (2) Each advertiser chooses one secret to mark packets and communicates it to all SPM-filters, who store it in parameter tables. Per source secret instead of per AS pair secret is necessary to ensure table scalability. (3)



(a) TP and RAP in isolated deployment.

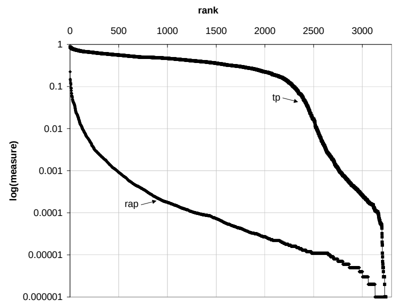


(b) TP in optimal deployment

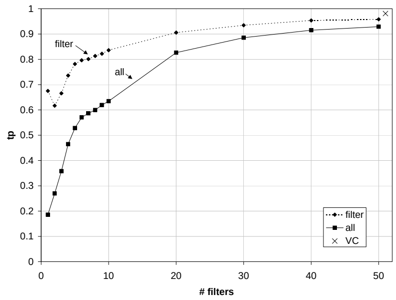


(c) RAP in optimal deployment

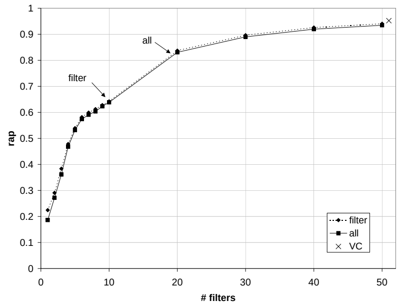
Figure 3: Performance measures for HCF



(a) TP and RAP in isolated deployment.

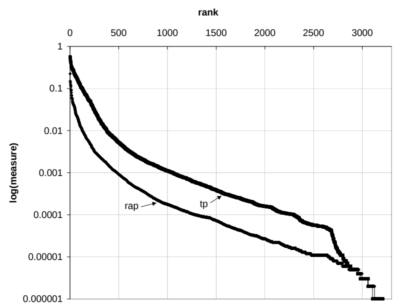


(b) TP in optimal deployment

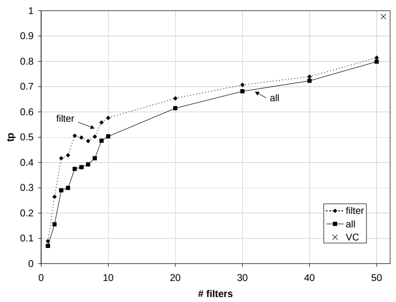


(c) RAP in optimal deployment

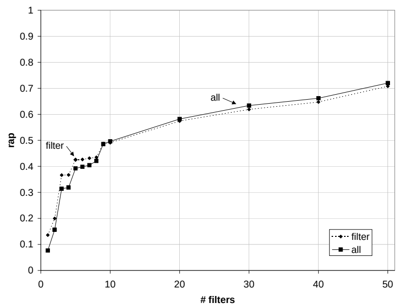
Figure 4: Performance measures for RBF



(a) TP and RAP in isolated deployment.

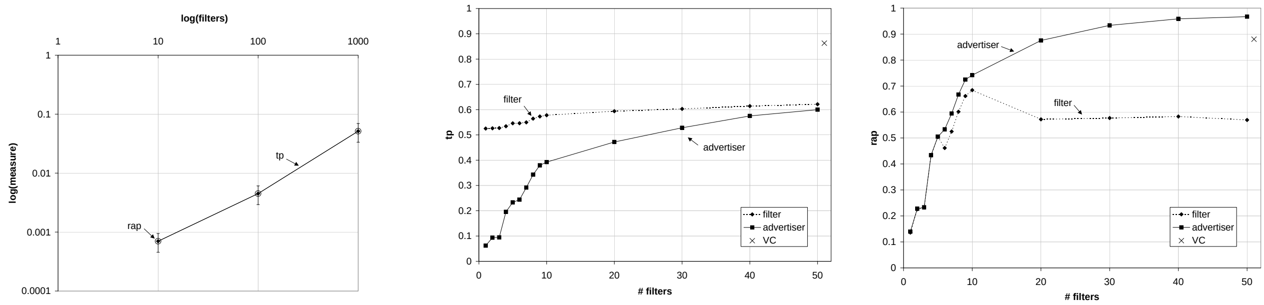


(b) TP in optimal deployment



(c) RAP in optimal deployment

Figure 5: Performance measures for IDPF



(a) TP and RAP in isolated deployment.

(b) TP in optimal deployment

(c) RAP in optimal deployment

Figure 6: Performance measures for SPM

Filters verify marks in packets from advertisers. Altruistic SPM would have to be properly secured, but we only focus here on evaluating its potential effectiveness. The advantage of altruistic vs. selfish SPM design can readily be grasped if we revisit the previous example in which A and B are SPM-advertisers, and some transit AS C is an SPM-filter. Spoofing of A's and B's addresses is now prevented in all traffic, regardless of the source, as long as it hits C. If C is chosen to be a popular node the benefit to A and B will be far greater than if they were deploying selfish SPM. Furthermore, selfish and altruistic SPM are complementary and can both be deployed to further increase spoofing protection.

For random, sparse deployment we choose 100 largest ASes as SPM-advertisers and show the strength of potential SPM-filters in Fig. 6(a). Because the strength of a filter only depends on $|IP_{spm}|$ it is constant and equal to 56%, and the filter impact follows the popularity trend (not shown). TP measures are lower than for HCF (Fig. 6(b)) and show an upward trend for optimal deployment, because of the added benefit of ingress filtering at popular nodes, that are also large. RAP measure is very high for advertisers because filters prevent spoofing of their addresses in all packets that hit a filter (Fig. 6(c)). Because some nodes are chosen both as filters and as advertisers, filters' RAP measure is higher than that of other nodes.

5.6 Packet Passports

Packet passports are attached by participating senders to their packets, and contain a sequence of marks where each mark is created using a secret shared between the source and one AS on the path to the destination. Participating routers verify marks and drop packets that fail this check. While packet passports have additional mechanisms to prevent replay attacks and sniffing, their basic design is similar to the design of the altruistic SPM. We thus omit graphs that express the effectiveness of packet passports.

5.7 StackPi

StackPi deploys distributed *markers* — routers that mark forwarded packets by shifting their IP identification field and appending a short label, which is derived from the previous hop's and this router's IPs. When the packet arrives at its destination, its mark consists of "stacked" labels placed by markers that have forwarded this packet, and can be used as "path identifier". The mark can be used for spoofed packet filtering by associating it with the source address, and dropping packets with incorrect marks. For this functionality, it is important to place the markers in such a way to maximize the distinctiveness between sources. This is achieved when stacking is minimized, i.e. when markers are as close to the sources as possible. SPi was proposed as a selfish defense, but the necessity of placing markers at remote routers makes it more suitable for altruistic deployment; marker placement must then maximize mark distinctiveness for all Internet nodes.

We select optimal markers using the following algorithm: (1) Sort ASes by 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.

Normalized strength of an SPi filter is:

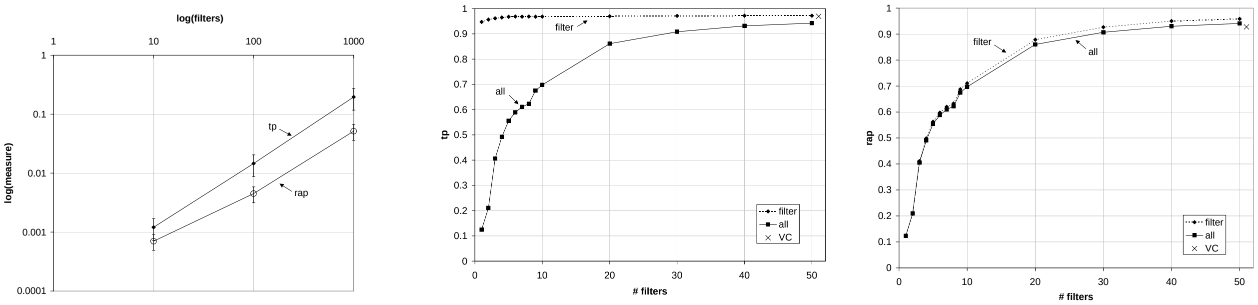
$$ns_F = \sum_{p \in IP_{allc}} Mark(p)/|IP_{allc}|^2, \qquad (18)$$

where $Mark(p)$ is the number sources whose mark at the filter differs from p's mark.

We evaluate SPi in random, sparse deployment following a strategy similar to the one we used for SPM evaluation. SPi's performance in this setting resembles performance of selfish SPM (Fig. 7(a)). In optimal deployment, TP and RAP measures (Fig. 7(b) and 7(c)) are very similar to HCF's, with filters receiving significant spoofed traffic protection.

5.8 Comparative analysis

Altruistic versions of HCF, RBF and SPi have a comparable performance across all measures and offer significant and fair protection to all nodes with 50 optimal filters. IDPF and SPM offer much lower protection with 50 optimal filters, while ingress filtering requires at least 10% deployment for 90% protection. SPi and SPM further need 100 optimally placed markers. HCF is the only defense that successfully protects filters from spoofed traffic in selfish deployment. RBF filters can benefit from selfish deployment, but only if they have a sufficient connectivity, while SPi filters require well-placed markers to reap benefits. Selfish deployment of IDPF only helps a few well-connected filters, while selfish SPM and selfish ingress filtering offer poor protection.



(a) TP and RAP in isolated deployment.
(b) TP in optimal deployment
(c) RAP in optimal deployment

Figure 7: Performance measures for SPi

6. REALITY OF EVALUATION SETTINGS

We have shown that the performance of all spoofing defenses depends on the filter popularity and strength. Topologies with a few very popular nodes facilitate efficient filtering with a small number of filters. Further strengths of HCF, RBF, IDPF and SPi filters also depend on connectivity and routing, because they associate source addresses with some route-dependent feature. Results obtained in our study are thus closely linked to the topology and routing inference approaches we used.

The current state of knowledge about the Internet's route map and forwarding behavior of different ASes necessitated a lot of educated guesses to fill the missing information. Yet, it begs the question "what if we made wrong guesses". Namely, what if the Internet's route map is such that evaluated defenses would have vastly different performance than in our study? We address this question by considering two worst-case scenarios: (1) A routing scenario which avoids popular nodes. Since popularity correlates strongly with connectivity, this rule should lead to route map in which large, well connected ASes are not popular, thus undermining their ability to act as good filters. We note that such scenario is unrealistic, for [17] has found that many routes travel on a few popular AS paths, implying that a few ASes are very popular. (2) Multipath, hot-potato forwarding that creates multiple routes between a source and a filter. This should lower the strength of route-dependent defenses — HCF, RBF, IDPF and SPi — because of the greater probability that allowed parameters for any two sources overlap allowing them to spoof each other. While we do not know how present hot-potato forwarding is in the Internet, we do know that ASes harbor multiple paths as shown in [17], so this scenario is realistic. Figures 8 and 9 show performance of SPM and HCF on a topology reconstructed from data we obtained from authors of [17]. The data contains the routing tables for a 1,000-node subset of the AS topology. This results in a $\approx 10,000$ node topology (inferred from routes), with complete routing information for 1,000 ASes.

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 our multipath topology. There are about 10,000 ASes in this 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 our multipath toplogy topology and 5% lower in the UCR topology. Thus, continued good performance of HCF does not stem from a bias present in the small multipath topology toward large filters, but from inherent topological properties (a few very popular paths, Gaussian hop count distribution) that exist even with multipath forwarding.

It may seem potentially useful to evaluate a third scenario — one in which the AS connectivity is increased to further decrease the likelihood of having popular nodes. However, a recent study [27] showed that AS topologies inferred from three available data sources — Skitter [19], RouteViews [13] and WHOIS [18] — are quantitatively but not qualitatively different. Assortiativity, which determines the prevalence of links between small-degree and large-degree nodes is very similar for all three topologies; the result of this is that many route inference strategies will produce routes that predominantly traverse a few popular, large-degree nodes. Out of three data sources, WHOIS data contains most links because it contains both currently used and backup links. If such a topology, with many obviously redundant links, has high assortiativity we have no reason to believe that the real Internet's topology does not.

7. CONCLUSIONS

Evaluation results speak strongly in favor of a systematic, Internet-wide deployment of spoofing defenses at strategically positioned ASes. For space reasons we could not elaborate on filter identities in the paper, but they are large, well-traversed, well-connected networks, that comprise the Internet's core. Only HCF offered significant protection in selfish, isolated deployment. Even then, deploying networks were protected against spoofed traffic but not against reflector attacks. SPM, SPi and HCF were much more effective as altruistic than as selfish defenses, and only altruistic defenses achieved sufficient path coverage to reduce reflector attacks. Optimally deployed filters dramatically outperformed random deployment in all scenarios, further supporting a case for strategic filter deployment. In case of two marking defenses, SPM and SPi, 100 core markers filtering in SPi achieved much higher effectiveness than 100 edge markers in SPM; this is one more argument in favor of community-based spoofing defenses. Prior research [7, 8] proposed vertex cover filter deployment for Internet-wide protection. Our results show that a much lower deployment at 50 chosen ASes can achieve a comparable effectiveness.

8. 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] R. Beverly and S. Bauer. The Spoofer Project: Inferring the Extent of Internet Source Address Filtering on the Internet . In *Proc. of SRUTI*, 2006.

[4] 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.

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

[6] 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.

[7] 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*, 2006.

[8] 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.

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

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

[11] 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.

[12] 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.

[13] University of Oregon. RouteViews Archive.

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

[15] L. Gao. On inferring autonomous system relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, November 2000.

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

[17] W. Mhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-Topology Model that Captures Route Diversity. In *Proc. of ACM SIGCOMM*, 2001.

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

[19] CAIDA. Skitter data.

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

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

[22] University of Michigan. Inet Topology Generator.

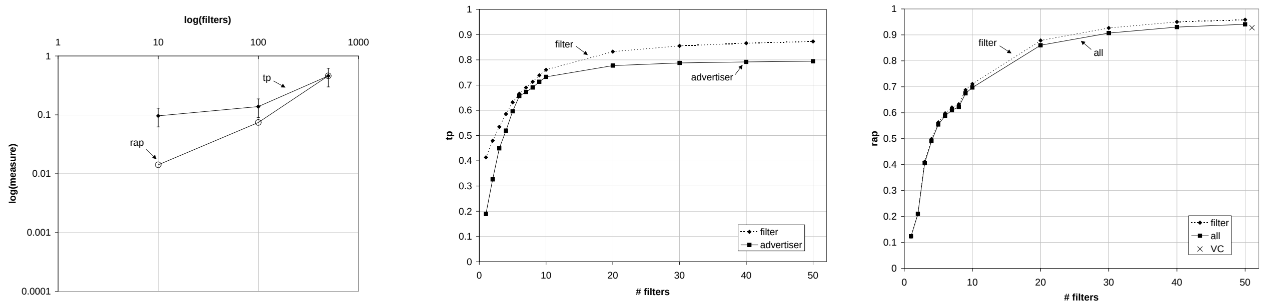
[23] G. Siganos and M. Faloutsos. Analyzing BGP Policies: Methodology and Tool. In *INFOCOM*, 2004.

[24] B. Cheswick. Internet Mapping project. `http://www.cheswick.com/ches/map/index.html`.

[25] Bogon IPs. `http://www.completewhois.com/bogons/`.

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

[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.

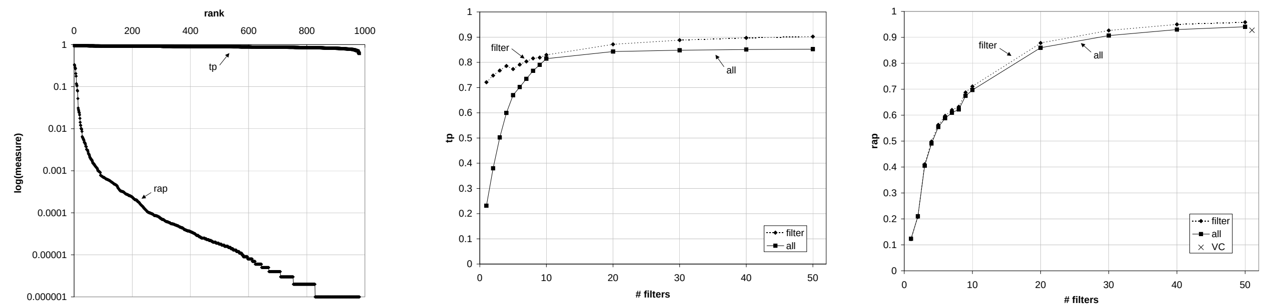


(a) TP and RAP in isolated deployment.

(b) TP in optimal deployment

(c) RAP in optimal deployment

Figure 8: Performance measures for SPM with multipath forwarding



(a) TP and RAP in isolated deployment.

(b) TP in optimal deployment

(c) RAP in optimal deployment

Figure 9: Performance measures for HCF with multipath forwarding