Towards A Cooperative Mechanism Based Distributed Source Address Filtering

Jie Li[*,1,2,3], Jun Bi[†,2,3], Jianping Wu[‡,1,2,3]
[1] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3] Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
[*] jieli@csnet1.cs.tsinghua.edu.cn, [†] junbi@tsinghua.edu.cn, [‡] jianping@cernet.edu.cn

***Abstract*—While making the Internet totally trustworthy is intractable, making as trustworthy as possible is a crucial problem. Within this landscape, authentication of the IP source address remains one important topic in need of further study. However, most source address validation methods are difficult to implement in practice because of deployment difficulties. This research designs an efficient inter-domain distributed source address validation solution (CatchIt). By employing a novel routing choice notification scheme, CatchIt makes the deployed ASes intelligent by allowing them cooperate to acquire the valid incoming path information of packets. With such knowledge, the deployed ASes can accurately authenticate the source address without the need for any modifications to the de facto routing protocol and packet structure. Moreover, CatchIt helps the deployed ASes proactively and quickly filter spoofed packets before they imperil the network. CatchIt also avoids any false positive, even under partial deployment. Our evaluation also shows that CatchIt is effective and accurate when catching spoofed packets while incurring a low overhead; CatchIt maintains an early deploy and rapidly benefit incremental deployment incentive mechanism.**

Keywords—IP source address validation; routing choice notification; inter-domain cooperation; network security

I. INTRODUCTION

The Internet has been subject to the IP spoofing threat for many years. This is a long-recognized consequence of the Internet's lack of verification of the IP source address. This functional vulnerability of the original design of the Internet provides a boon for an attacker to launch a wide array of IP spoofing attacks against legitimate users. The IP spoofing attacker disguises its true origin by inscribing bogus information in the source address field. The aim of these attacks is to disrupt the normal operation of the targeted network system by depleting its resources and render any source-based service unavailable. Due to the hardness of pinpointing the true origin of an attacker, IP spoofing is still a popular attack vector. Research [1] shows that spoofing-based attacks induce financial losses ranging from hundreds of thousands to millions of dollars per hour. The largest reported spoofing-based attack (DDoS) size doubled over the one in 2009, which reached 100 Gbps in 2010 [1].

Even if attackers can insert forged source addresses into IP packets, they cannot control the actual paths that the packets take to the destination. Using route-based path authentication, therefore, is a clever way of conducting source address validation and combating IP spoofing attacks. If we equip many routers with trustworthy validation rules specifying valid incoming interfaces for source prefix spaces, we can thwart an attacker's malicious behavior by reducing their available forgeable source prefix spaces. Unfortunately, current solutions [3], [4] are difficult to deploy in large scale networks because of the intractability of building an accurate route-based packet filtering due to the lack of inter-domain cooperation. [5] uses route constraints in networks to determine whether a packet, given its source and destination address, is misrepresenting its true origin. However, the system requires full deployment in a network, which is difficult to do in real world networks.

In this paper, we propose a novel distributed source address validation solution with inter-domain ***C***ooper***A***tive rou***T***ing ***CH***oice notif***I***ca***T***ion mechanism (*CatchIt*). CatchIt is crafted to utilize a novel route-based path authentication logic mechanism that mitigates IP spoofing based on the collaborative efforts of only a subset of AS Border Routers (ASBR) on the Internet. By employing an intelligent routing choice notification scheme, CatchIt achieves inter-domain routing system cooperation and enforces that deployed ASBRs build a more accurate filtering table (used as validation rules) to discard spoofed packets while at the same time allowing legitimate ones to reach the destination.

CatchIt offers three desirable benefits.(1) Proactive and fast defense: CatchIt can not only successfully filter spoofed packets before they jeopardize networks, but also filters out the packet as near to the malicious source as possible instead of passively defending at the destination; (2) Efficiency and trustworthiness: CatchIt offers gains in efficiency by making the deployed ASes intelligent and cooperative. CatchIt facilitates the process of accurately verifying the source address and dynamically constructing spoofed packet filters all while avoiding any false positives; (3) Deployment incentives: CatchIt enables deployed ASes the ability to protect their internal network from receiving spoofed packets even under partial deployment, without the need for any modifications to the de facto routing protocol and packet structure. CatchIt also can provide additional benefits to ASes when they deploy the solution. CatchIt maintains an *early deploy and rapidly benefit* incremental deployment incentive mechanism.

The rest of this paper is organized as follows: Section II describes the design principle underlying CatchIt in detail; Section III evaluates CatchIt's efficacy, correctness, and overhead; Section IV discusses related work; and finally, Section V concludes the paper. For the remainder of this paper *deployed AS* will refer to *CatchIt-enabled AS*, *deployed ASBR* will refer to *CatchIt-enabled ASBR* unless otherwise noted.

978-1-4673-5775-3/13/$31.00 ©2013 IEEE

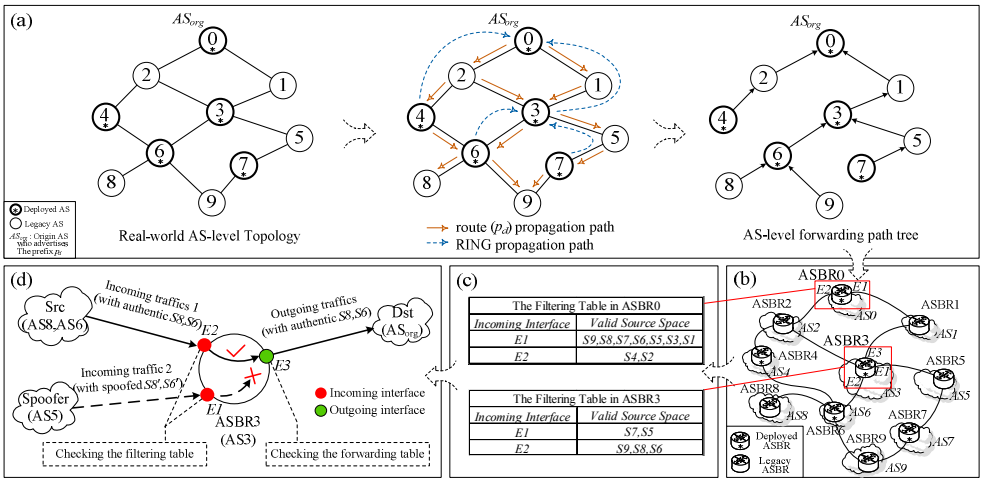

Fig. 1(a)-(d). Demonstration of the methodology of the RING-based distributed inter-domain source address validation solution (CatchIt).

II. DESIGNING CATCHIT

Although outgoing interfaces are recorded in every router's forwarding table, the incoming interface knowledge is yet to be discovered. The goal of CatchIt is to ensure that each deployed ASBR can acquire valid incoming path information of packets carrying a specific source prefix, even under partial CatchIt deployment. Thus, each deployed ASBR can utilize the information to build a filtering table to filter spoofed packets.

A. Basic Setting

We identify basic assumptions that motivate our design presently. (1) Deployed ASes can obtain the mapping between a source prefix and the corresponding origin AS. This may be provided by security mechanisms as suggested in some IETF standards (e.g., RFC 6480 and 6482). With this assumption, a deployed AS can obtain the accurate mapping information between a source address and the corresponding source AS from BGP using the AS_PATH path attribute. (2) Only end hosts attempt attacks on the network; ASBRs are not malicious. Without this assumption, malicious ASBRs can bypass our protection perimeter. Thus, malicious ASBRs can wreak havoc on the Internet from the inside.

Given any reachable prefix p_d, AS_{org} refers to the origin AS who advertises p_d. Given any two deployed ASes, *ASi* and *ASj*, if packets originating from *ASi* transit *ASj* en route to destination p_d without transiting any other deployed AS on the actual path, *ASj* is referred to as an *upstream logical neighbor* of *ASi*. *ASi* is a *downstream logical neighbor* of *ASj* accordingly. Given any two ASes, *ASu* and *ASd*, if *ASu* directly connects to *ASd* through a physical link, *ASu* and *ASd* are referred to as *physical neighbors* of one another.

B. Design Rationale

Each AS in today's Internet uses the BGP routing protocol to decide the best forwarding path from every source prefix to every destination prefix. That is, each AS only determines the outgoing interface for every destination. However, there is no facility available for each AS to learn which incoming interface a packet with a specific source prefix should come from. Without introducing any modifications to the de facto BGP protocol and packet structure, CatchIt adopts an intelligent ***R***outing cho***I***ce notificatio***N*** messa***G***e (RING). RING is designed to inform upstream deployed ASes about the actual path that has already been chosen by downstream ASes (deployed ASes and legacy ASes) towards a given destination p_d. This enables deployed ASes to learn the valid incoming interface of packets with a specific source prefix space while at the same time verifying whether each packet arrives at the expected interface on the actual path. Equipped with this routing choice notification scheme, CatchIt not only makes the control plane of deployed ASBR intelligent and cooperative, but also enhances the ability of the data plane to accurately validate the source of a packet to the granularity of the origin AS, even with the presence of legacy routers. The demonstration of the methodology of CatchIt is shown in Fig.1.

C. Routing Choice Notification Message

As discussed in Section II-B, deployed ASes exchange RING messages based on the collaborative efforts of only a subset of deployed ASBR in the network. To do so, CatchIt employs a special globally unique AS Identification, namely AI. AI is a specified globally routable IP address that indicates the interconnected point that helps deployed ASes set up a logical *inter-domain cooperation channel* through an established secure TCP connection. CatchIt piggybacks AI in a BGP update and sets the path attribute of this BGP update to optional transitive. This provides backward compatibility for enabling legacy ASes to forward the AI information in an uninterrupted fashion as shown in left part of Fig.2.

CatchIt requires that RING contains three essential fields. The *destination prefix space field* (*DPS*) specifies a reachable destination prefix carried in a selected route after BGP decision process. The *source prefix space field* (*SPS*) keeps a record of source prefix spaces on the actual path along which packets originating from these recorded source prefix spaces traverse the upstream logical neighbor AS toward the destination indicated on DPS. Each RING will cross a series of deployed ASBRs, each of which will update its incoming path information based on the SPS contained in this RING (incoming path information acquisition is discussed further in Section II-D). The SPS itself will also be updated in transit, as illustrated later. The *interconnected point field* (*ICP*) indicates

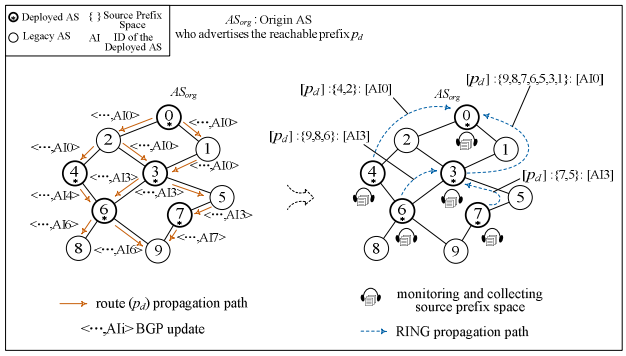

Fig. 2. Routing Choice Notification Message Propagation.

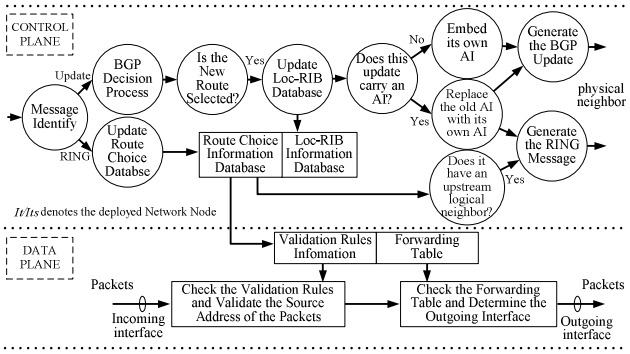

Fig. 3. Implementation of CatchIt in Deployed Network Node.

the AI of the upstream logical neighbor. The main form of the RING is [DPS]:{SPS}:[ICP] as shown in right part of Fig.2.

D. Acquiring Incoming Path Information

As depicted in Fig.3, the main components of the CatchIt implementation are generating RINGs, processing RINGs, performing the routing choice notification, generating filtering table and implementing validation. We will describe each of these operations.

1) AS-level Forwarding Path Tree

The routing choice notification procedure begins when a deployed ASBR *ASBRi* residing in a deployed AS *ASi* receives a route $r(p_d)$ (carrying the prefix p_d) from a physical neighbor AS. If *ASBRi* selects $r(p_d)$ as the actual route toward the destination p_d after the BGP routing decision, then *ASBRi* updates its Loc-RIB and identifies whether there exists an upstream logical neighbor *ASj* thus continuing the routing choice notification process with *ASj*.

In order to identify whether there exists such an *ASj*, *ASBRi* checks the received BGP update $u(r(p_d))$ (carrying the selected $r(p_d)$) to identify whether it carries an AI *AIj* that belongs to the *ASj*. If *ASBRi* identifies that the $u(r(p_d))$ *does* carry an *AIj*, then there exists an upstream logical neighbor *ASj* of *ASi*. That is, packets originating from *ASi* will transit *ASj* en route to destination p_d. *ASBRi* will then make a record of *AIj* and replace *AIj* with *ASi*'s AI *AIi* that is embedded into *ASBRi's* new BGP update. As depicted in left part of Fig.2, when the deployed AS *AS4* selects the received route $r(p_d)$ (carried in the BGP update <…,AI0>) from a physical neighbor AS *AS2*, *AS4* simultaneously identifies that there exists an upstream logical neighbor AS *AS0* by analyzing <…,AI0>. *AS4* will then make a record of AI0 and replace AI0 with AI4 that is embedded into the new BGP update <…,AI4>. If *ASBRi* identifies that the $u(r(p_d))$ *does not* carry an *AIj*, then there does not exist an upstream logical neighbor *ASj* of *ASi*. In this case, *ASBRi* just embeds *AIi* into its new update.

ASBRi can connect and communicate the RING to *ASj* by retrieving the record of *AIj*. Meanwhile, *ASi*'s AI *AIi* is piggybacked in *ASBRi's* new BGP update and is delivered to the next hop. This assists downstream logical neighbors in establishing the inter-domain cooperation channel with *ASi*. *ASBRi* collects the source prefix spaces of upstream legacy ASes on the path between *ASi* and *ASj* by analyzing the AS_PATH information. *ASBRi* also dynamically monitors the source prefix spaces belonging to downstream ASes whose packets transit *ASi* heading for p_d.

After connecting the upstream logical neighbor *ASj*, *ASBRi* starts performing the RING generation. *ASBRi* firstly sets up the DPS as p_d, then sets up the ICP as *AIj*, and lastly inserts the source prefix spaces into the SPS. These source prefix spaces consist of three parts: the *ASi*'s source prefix spaces, the source prefix spaces belonging to upstream legacy ASes on the path between *ASi* and *ASj*, and the source prefix spaces belonging to downstream ASes whose packets transit *ASi* en route to p_d.

The generated *ASi*'s RING is forwarded along the same actual path as the packets from these prefix spaces. This informs the upstream logical neighbor *ASj* of the valid incoming interface for these source prefix spaces. Upon receipt of *ASi*'s RING, *ASj* splits the SPS in *ASi*'s RING, then generates and sends its own RING to its upstream logical neighbor. In this fashion, each deployed AS forwards its own RING to its upstream logical neighbor one by one, until the most upstream deployed AS who does not have an upstream logical neighbor receives its downstream logical neighbor's RING. During this RING propagation process, each *ASBRi* in transit will update its mapping between the source prefix spaces and the corresponding incoming interface.

The whole process of propagating RING is illustrated in right part of Fig.2. In order to inform upstream deployed AS *AS0* about the actual path that has already been chosen by downstream ASes (deployed ASes *AS6 AS7* and legacy ASes *AS8 AS9*) towards the given destination p_d, a deployed AS *AS3* communicates its RING to the upstream logical neighbor *AS0*. *AS3*'s RING not only informs the actual path, but also records all source prefix spaces belonging to the ASes on the actual path. *AS3* inserts three parts source prefix spaces into its RING, its own source prefix space {*S3*}, the source prefix space {*S1*} belonging to upstream legacy AS *AS1* on the path between *AS3* and *AS0*, and the source prefix spaces {*S9*, *S8*, *S7*, *S6*, *S5*} belonging to downstream ASes (*AS9*, *AS8*, *AS7*, *AS6*, *AS5*) whose packets transit *AS3* en route to p_d.

In this way, CatchIt finally builds a logical directed *AS-level forwarding path tree* based on RING in a distributed fashion. Fig.1(a) shows a case of AS_{org}-root AS-level forwarding path tree. In the AS-level forwarding path tree, each AS node represents a specific source prefix space and is associated with a specific incoming interface. Each branch represents the actual path along which packets originating from downstream AS nodes take to destination p_d. The AS-level forwarding path tree is stored in the routing choice information database as shown in Fig.3. This tree is used to derive the filtering table. Triggered by both routing table changes, CatchIt

TABLE I. STATISTICS OF TOPOLOGIES USED IN EXPERIMENT I,II

Topology	Properties of The Four Graphs			
	Data Sources	*Collection Date*	*Fidelity (%)*	*Correlation Coefficient*
INET311	Map120701	07.01.2012	98.12%	0.9839
INET400	Map120527	05.27.2012	97.53%	0.9451
INET525	Map120701	07.01.2012	98.77%	0.9878
INET610	Map120501	05.01.2012	99.42%	0.9901

TABLE II. STATISTICS OF TOPOLOGY USED IN EXPERIMENT III

Member	Statistics Data		
	Distribution	*# of ASes*	*Percentage (%)*
CERNET2	China	25	11.1%
ChinaTelecom	China	45+	20%
ChinaMobile	China	55+	24.5%
TEIN3	Trans-Eurasia	20+	8.9%
GÉANT2	Europe	34+	15.1%
APAN-JP	Japan	32+	14.17%
KREONet2	Korea	14+	6.23%
Total	/	225+	100%

periodically initiates updates in order to reconstruct the forwarding path tree and incoming interface mapping.

2) Generating and Updating Filtering Table

When each *ASBRi* receives the RING, the RING will be correctly retrieved. Each *ASBRi* can record the path that the RING has traversed up until that point and ensures that its own RING follows the same path toward the destination p_d the same way as valid data packets do. The RING is further processed in order to help *ASBRi* build a route-based filtering table so as to ensure the source prefix spaces are mapped to the incoming interface in a distributed fashion as shown in Fig.1(b,c). On the AS-level forwarding path tree, once an AS node's parent is changed, both its new and old parent will automatically modify their incoming interfaces so as to remap all descendant source prefix spaces related to the incoming interfaces.

3) Processing the Routing Choice Notification Message

CatchIt enables aggregation in RING for reducing bandwidth consumption. That is, each *ASj* not only forwards its own RING, but also aggregates downstream logical neighbor *ASi*'s RING by updating the SPS field while in transit. We set a timer to enable an efficient piggyback mechanism so that the RINGs can aggregate along the route as much as possible.

E. Implementing Validation

CatchIt adopts the filtering table as validation rules to verify if a packet arriving at an *ASBRi* is authentic in light of the mapping between the source prefix spaces and the expected incoming interface. CatchIt identifies and discards any forged packets. An example case is illustrated in Fig.1(d). It can clearly be seen that packets originating from *AS8* with authentic source *S8* are transmitted to a user h_d located in AS_{org} along the actual path *path*<*AS8*,*AS6*,*AS3*,*AS1*,*AS0*> and arrive at the expected incoming interface *E2* in *ASBR3*. Suppose an attacker at *AS5* is attempting to send a number of malicious packets targeted at the same user h_d with forged source *S8'* belonging to *AS8* along the actual path *path*<*AS5*,*AS3*,*AS1*,*AS0*>. When these forged packets are routed to *ASBR3* with respect to the incoming interface *E1*, *ASBR3* will validate the authenticity of source addresses of these forged packets by checking its filtering table. Due to the mismatch between the legitimate path and incoming interface, *ASBR3* is able to discern that the source addresses of these packets must be spoofed. Thus, *ASBR3* discards these spoofed packets while allowing the legitimate ones to reach h_d. In this way, *ASBR3* proactively protects the h_d and *AS8* from the spoofing attack as close to the malicious source *AS5* as possible as opposed to passively defending at destination p_d.

F. Security and Feasibility Concerns

For the sake of preventing eavesdropping and the interception of validation information, we specially take account of the following two secure designs schemes:

(1) The secure channel scheme: CatchIt takes special care to secure the RING communication against malicious attempts to compromise, misuse or disable the protocol. In order to provide the safe delivery of data, deployed ASBRs can utilize a combination of TCP intercept and security protocol suites to set up a secure channel for the safe communication of RING. Examples of such suites include the Diffie-Hellman public-private key agreement protocol or IPSec.

(2) Loose coupling scheme: CatchIt allows deployed ASes to implement simultaneously different levels of granularity of source address validation at the access network level and intra-AS level, respectively. The coupling of components at different levels of granularity of authenticity is loose enough to allow component substitution.

In order to enhance the feasibility of deploying filtering ASBRs that achieves a high degree of filtering effectiveness with a small number of deployment points, we also concentrate on ensuring CatchIt's deployment feasibly in the real world:

(3) Flexible placement strategy: Previous studies [2] found that a vertex cover of a filtering system would be perfectly effective, but such a deployment would be extremely difficult if not impossible to actually implement, especially in an AS-level topology. We achieve flexible placement with a random deployment strategy. We randomly arrange the ASes to deploy CatchIt on the AS-level network topology. Experimental results show that CatchIt can effectively limit IP spoofing attacks using a flexible placement strategy even with a low percentage of deployment (Section III).

(4) Additional benefits: By employing the inter-domain cooperative routing choice notification mechanism, CatchIt gives deployed ASBRs the valid incoming path knowledge. This allows for additional benefits other source address validation methods cannot offer. For instance, reverse path forwarding, used in some implementations of IP multicast, could easily acquire the valid reverse path. CatchIt not only protects a CatchIt-protected network from reflection attacks, but also it protects it from misplaced blame. Meanwhile, CatchIt allows a deployed AS provide source validation as a charged service to other ASes who concerns IP spoofing and would like to buy this service. The more services a deployed AS can provide, the more profit it can make, and by extension the more potential ASes CatchIt can attract to deploy itself.

III. ANALYSIS AND EVALUATION

A. Formulation and Objectives

One must make proper measurements in order to understand the performance of any system. We will evaluate

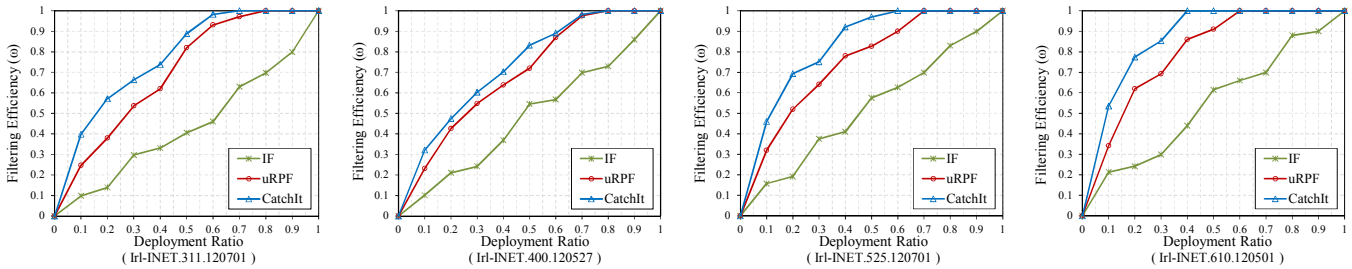

Fig. 4. CatchIt validation effectiveness evaluation (filtering efficiency on the 4 groups of As-level topologies).

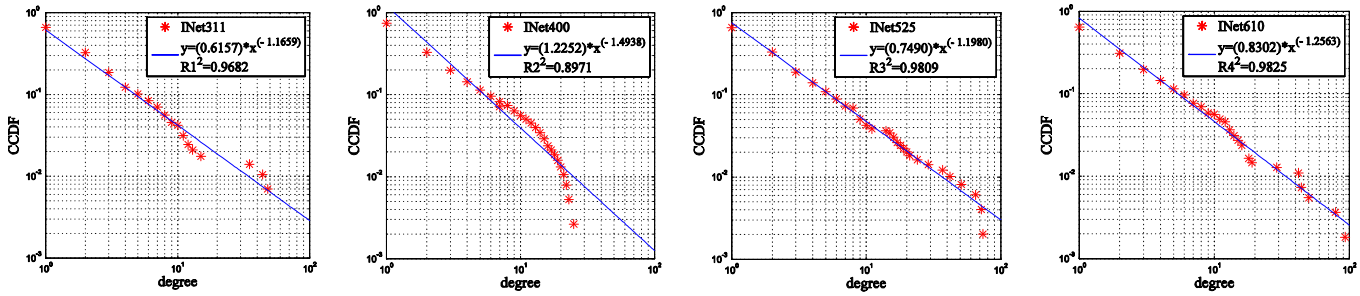

Fig. 5. CCDF measurement of the 4 groups of AS-level topologies (Log-log plot).

the advantages of CatchIt on three crucial metrics: *filtering efficiency* (how well CatchIt is able to filter spoofed packets), *false positive* (how often CatchIt mistakenly drops legitimate packets) and *storage overhead* (how much space CatchIt's filtering table requires). Recall [3,4] are typical path authentication based methods and both support incremental deployment. Their mechanisms is similar to CatchIt's. Compared with [3,4], we show how more effective and accurate CatchIt is in filtering the spoofed packets.

We let D refer to the set of deployed ASes, *src* refer to the set of all Internet source prefix spaces and *dst* refer to the set of all reachable destination spaces, respectively. Let *spf* be the set of spoofed source prefix spaces, and $\overline{spf}$ be the set of authentic source prefix spaces. $\forall ASi \in D$, ASk is defined as the downstream AS of ASi. Given a destination $d \in dst$ belonging to AS ASm, P_i<d,s> with the source s refers to the packet originating from ASk that transits ASi heading for d residing in ASm along the path $path$<ASk,…,ASi,…,ASm>. Let R represent the set of ASBRs residing in ASi, and $fl_{rj}^{(i)}(P_i < d, s >)$ be the filtering function deployed on ASBR rj residing in ASi ($i \in |D|, j \in |R|$). For each deployed AS ASi, we let AR refer to the overall filtered attack traffic (attack traffic targeted at ASi and attack traffics targeted at other ASes) that reach ASi. This AR is defined as follows:

$$AR = \bigcup_{j \in |R|} \bigcup_{\substack{s \in spf \\ d \in dst}} fl_{rj}^{(i)}(P_i < d, s >) \qquad (i \in |D|) \quad (1)$$

We let AW refer to the overall filtered legitimate traffic that is mistakenly dropped by the defense mechanism in ASi. The AW is defined as follows:

$$AW = \bigcup_{j \in |R|} \bigcup_{\substack{s \in \overline{spf} \\ d \in dst}} fl_{rj}^{(i)}(P_i < d, s >) \qquad (i \in |D|) \quad (2)$$

B. Experimental Settings

We use SSFNet [7] to implement CatchIt under the precondition that validation rules have already been constructed. We simulated BGP [6] for inter-domain routing. We also introduced asymmetric routing. The simulation implements the routing policy recommended by [16]. Our simulations run on the designed three experiments described in Section III-C. We randomly deploy CatchIt-enabled ASes throughout the network, and vary the deployment percentage between 0% and 100% in 10% increments. Experimental results show how the metrics from Section III-A vary as more ASes begin to deploy CatchIt. We introduce routing changes in order to measure how effective and efficient CatchIt is at dropping spoofed packets and in ensuring that legitimate packets are not mistakenly marked as spoofed.

C. Experimental Evaluation

1) Experiment I. Filtering Efficiency

In Experiment I, we construct four AS-level topologies based on the Internet AS-level topology data archived by the UCLA Internet Topology Collection Project [15]. We select three Internet AS-level topologies Map120501, Map120527 and Map120701. Then, we utilize algorithm [8] to rebuild these three topologies to four AS-level topologies while still maintaining their essential characteristics. Finally, the four AS-level topologies, denoted INET311, INET400, INET525 and INET610 closely match real network characteristics, not just in terms of graph structure (node interconnectivity) but also with respect to various AS memberships and link latencies. More importantly, we also measure and plot the Complementary Cumulative Distribution Function (CCDF) curves of the four topologies on the log-log plot as shown in Fig.5. After applying a nonlinear fitting process, we capture critical information about the trend of power-law distributions on each of the four topologies. In Table I we summarizes the properties of the four topologies, including the data sources, collection date, fidelity and correlation coefficient (R).

To quantify and measure the filtering capability of CatchIt, we use the filtering efficiency metric to evaluate how CatchIt can improve validation effectiveness by filtering the spoofed

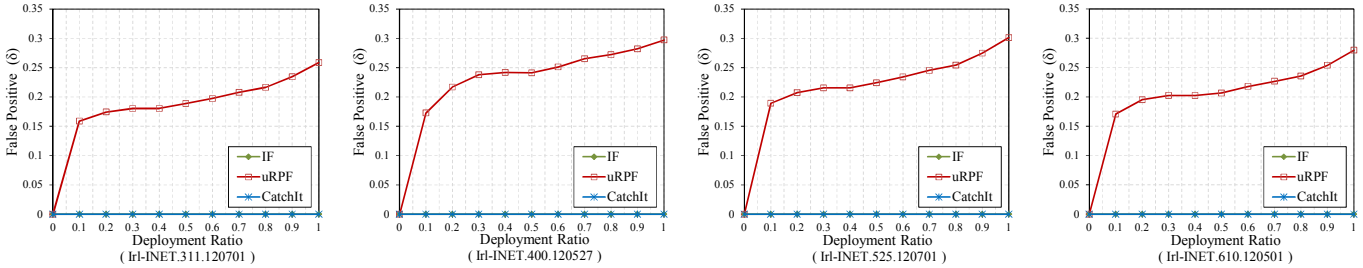

Fig. 6. CatchIt validation correctness evaluation (false positive on 4 groups of AS-level topologies).

packets, i.e. those packets captured by the overall spoofed packets filtering with rate ω. ω is defined as follows:

$$\omega = |\bigcup_{j\in|R|}\bigcup_{\substack{s\in spf\\ d\in dst}} fl_{rj}^{(i)}(P_i < d,s >)| / |\bigcup_{\substack{s\in src\\ d\in dst}} P_i < d,s >|,(i\in|D|) \quad (3)$$

We evaluate CatchIt on four different scenarios with varied deployment percentages according to Equa.(3) and measure the CCDF of each of the 4 scenarios as shown in Fig.5. Fig.4 shows that CatchIt can effectively limit the spoofing capability of an attacker even with a low percentage of deployment, e.g. attackers in a maximum of 85% (minimum 62%) of ASes cannot successfully launch a spoofing-based attack with only 30% CatchIt coverage on the network. Combining results from Figures 4 and 5, we see that the more the network topology adheres to the power-law (according to the studies in [17], the CCDF approximates to a straight line on the Log-log plot), the more significant attack prevention CatchIt can achieve, even at a low deployment percentages. For instance, at some random point in the network with only 20% CatchIt deployment, Fig.4 and Fig.5 show that ω reaches 78% on the typical power-law network INET610. More importantly, ω rapidly increases as the deployment percentage increases.

2) Experiment II. False Positive

We adopt the same data sets as described in Experiment I.

Another crucial metric we use is counting the number of false positives. Any desirable and practical packet filtering mechanism should focus on not dropping any legitimate packets, while swiftly dropping any detected malicious packets. Accordingly, we introduce the false positive metric as a correctness evaluation of CatchIt by defining rate δ.

$$\delta = |\bigcup_{j\in|R|}\bigcup_{\substack{s\in \overline{spf}\\ d\in dst}} fl_{rj}^{(i)}(P_i < d,s >)| / |\bigcup_{\substack{s\in src\\ d\in dst}} P_i < d,s >|,(i\in|D|) \quad (4)$$

We give the results of our evaluation of the false positive according to Equa.(4) in Fig.6. Fig.6 shows that CatchIt can effectively implement filtering and does not cause any false positives with an increasing deployment ratio. Although [3] is also effective at filtering, it has a negative effect on network operation (causing serious false positives as shown in Fig.6), allowing the filter to drop legitimate packets. Dropping legitimate packets is unacceptable. Although [4] also does not introduce any false positive which is zero as shown in Fig.6, filtering efficiency is weaker than CatchIt. In this sense, CatchIt is superior to the approaches used in [3] and [4] with regards to filtering efficiency.

In order to introduce routing changes, we randomly disconnect seven deployed ASes from the network. We then re-evaluate the filtering efficiency and the false positive. After

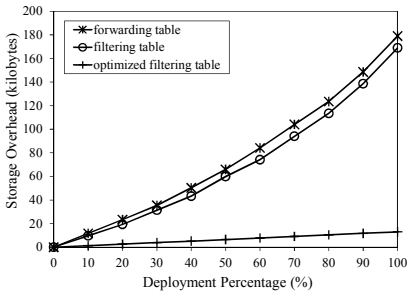

Fig. 7. CatchIt storage overhead evaluation.

that, these two crucial metrics by CatchIt remain essentially unchanged. The details of the results are omitted for brevity.

3) Experiment III. Storage Overhead

So far, for the goal to implement a trustworthy Internet infrastructure, security services and applications, the prototype system based on authenticated IPv6 source address validation architecture has been deployed on China Next Generation Internet (CNGI) [10]. Many research institutes and ISPs have become involved in this project, domestic members including CERNET2 [10], ChinaTelecom and ChinaMobile, overseas members including TEIN3 [11] GÉANT2 [12], APAN-JP [13] and KREONet2 [14]. These members are designated with more than 225 globally unique AS Nums, which formed the global Next Generation Internet (NGI). The global NGI provided an ideal network environment for Experiment III. Table II summarizes the partial properties of the global NGI.

The filtering table is not only built from the AS-level forwarding path tree, but is also built to be faster and more compact than the AS-level forwarding path tree. Recall ASBRs are equipped with highly optimized schemes for fast table lookup. In cases where the incoming interface from a specific source prefix space is the same as the outgoing interface to that source, CatchIt can implement redundancy optimization to reduce the storage overhead for the filtering table by leveraging symmetries in network routing. That is, a deployed ASBR does not need to generate and store filtering table entries for every source prefix space, a deployed ASBR only focuses on creating and storing entries for those source prefix spaces whose incoming interface are different from the outgoing interface. In this fashion, the forwarding table entry that corresponds with this source prefix space can be utilized to derive the valid incoming interface and thus validate the source address. If necessary, a flag can be added to the forwarding table entry to indicate that the deployed ASBR must consult the filtering table to determine the valid incoming interface. The degree to

which this optimization reduces storage overhead depends on the degree of asymmetry present.

Based on measured data collected from the Network Operation Center (NOC) of CNGI-CERNET2, we observe that about 87% of the total pairs of routes are symmetric in the CNGI-CERNET2. This implies that the CNGI-CERNET2 dataset only contains about 13% pairs of routes that have some degree of asymmetry at the AS level (i.e., the forward path and the reverse path pass through at least one different AS). Fig.7 shows that for reasonable cases in the CNGI-CERNET2, the storage overhead of the optimized filtering table can be minimal. The rate of optimization can reach to 87%. Comprehensive evaluations and experimental results both show that CatchIt is more effective and trustworthy than previous methods [3,4] and prove that the designed scheme is incrementally deployable.

IV. Related Work

For many years the research community has been committed to combating IP spoofing. Using path authentication is a practical way of defending against spoofing packets. Typical examples include [2],[3],[4],[5] and [9].

Route-based distributed packet filtering (DPF) [2] was the first effort to evaluate the relationship between topology and the effectiveness of route-based packet filtering. DPF studied how packet filters that are built based on the global routing information can significantly mitigate IP spoofing. However, DPF worked under the essential assumption that the filtering system already existed without actually designing any methods for routers to acquire the correct route for every source address. DPF does not provide direct incentives to deployers.

Inter domain packet filter (IDPF) [9] relies on the No-Valley-Customer-Prefer approach to locally infer a set of feasible routes instead of one best route. IDPF filters packets based on these feasible routes. IDPF only infers feasible paths, not actual paths. Although IDPF is easy to deploy, the local inference has the limitation of incompleteness. This makes IDPF slightly less accurate in filtering spoofed packets and partly limits IDPF's performance. In this paper, we improve the accuracy of filtering by running an inter domain cooperative mechanism. This helps ASBRs accurately learn the incoming path knowledge.

Unicast reverse path forwarding (uRPF) [3] requires that a packet is forwarded only when the incoming interface that the packet arrives on is exactly the same as the outgoing interface that the router used to reach the source IP of the packet. If the interface does not match, the packet is dropped. Although uRPF is a simple scheme, Internet routing is inherently asymmetric. Thus, uRPF often incurs accidental filtering of legitimate traffic.

Ingress filtering (IF) [4] was designed to make sure that incoming packets are actually from the networks that they claim to be from. IF effectively provides a special purpose incoming table only at network ingress. However, studies have shown that unless IF is almost full deployed, nearly arbitrary forgery is still possible. Further, this method offers no help in providing address assurance for any other purposes.

Source address validity enforcement (SAVE) [5] enables routers to build incoming tables that properly describe the mapping between incoming interfaces and source addresses. The tables are built using methods similar to the generation of routing tables. To some extent, SAVE works properly only when ubiquitously deployed at all routers. Such a requirement is a little bit unrealistic.

V. Conclusion

Our new solution, CatchIt, is a novel route-based path authentication methodology for mitigating IP spoofing based on the collaborative efforts of only a subset of ASBRs on the Internet. The major contribution of our work is our suggestion of a plausible way of making the Internet more trustworthy by enabling inter-domain routing system cooperation via an intelligent routing choice notification mechanism. The usage of this architectural enhancement offers a way to facilitate CatchIt to provide an accurate AS-level source validation service in an incrementally deployable way. In the future, we will perform a large-scale deployment in the CNGI and study the pricing and billing model and relative operation mechanisms, which describes how a deployed AS provides source address validation as a charged service to other ASes.

Acknoledgment

This work was supported by the National High-tech R&D Program of China under Grant 2013AA010605, the National Science Foundation of China under Grant 61073172, and the National Key Basic Research Program of China under Grant 2009CB320501. Jun Bi is the corresponding author.

References

[1] "Network Infrastructure Security Report," Arbor Networks. Available: http://www.arbornetworks.com/report. Feb., 2011.

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

[3] "Understanding Unicast Reverse Path Forwarding," Cisco Systems, Available: http://www.cisco.com,2012.

[4] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," RFC 3704, Mar 2004.

[5] J. Li, J. Mirkovic, M. Wang, et al., "SAVE: Source address validity enforcement protocol," in *Proc. IEEE INFOCOM*, New York,2002.

[6] Y. Rekhter, T. Li, and S. Hares,"A Border Gateway Protocol 4 (BGP-4)," RFC 4271,January,2006.

[7] SSF Network Models, Available: http://www.ssfnet.org/homePage.html.

[8] P. Mahadevan, C. Hubble, B. Huffaker, et al., "Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies," *ACM SIGCOMM*,37(4), 2007.

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

[10] CNGI-CERNET2, Available: http://www.cernet2.edu.cn.

[11] TEIN3, Available: http://www.tein3.net.

[12] GÉANT2, Available: http://www.geant2.net.

[13] APAN-JP, Available: http://www.jp.apan.net/NOC/.

[14] KREONet2, Available: http://noc.kreonet.net.

[15] "Internet Topology Collection", UCLA Computer Science Department's Internet Research Lab (IRL), Available: http://irl.cs.ucla.edu/topology/.

[16] S Halabi, "Internet Routing Architectures", Cisco Press,August, 2000.

[17] G. Siganos, M. Faloutsos, P. Faloutsos, et al.,"Power laws and the AS-level internet topology," *IEEE/ACM Transactions on Networking*, 2003.