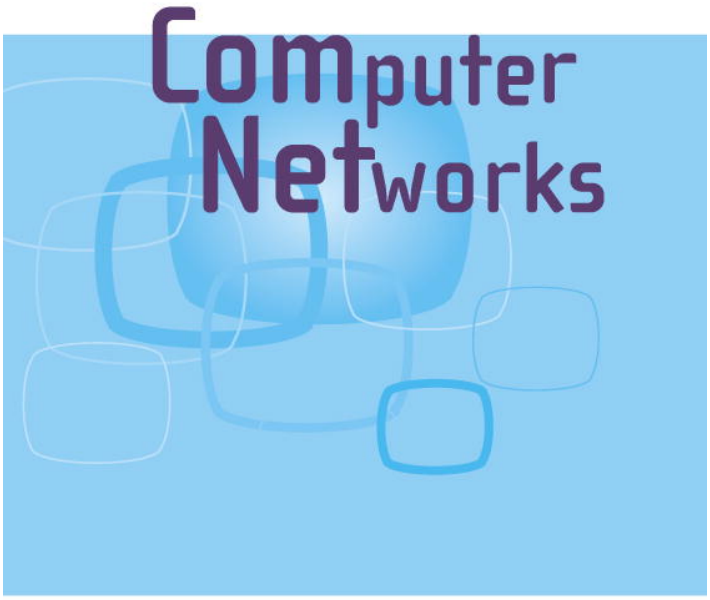
Volume 51 Issue 17 5 December 2007 ISSN 1389-1286



Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 4780–4796
Computer Networks
www.elsevier.com/locate/comnet

A measurement study of persistent forwarding loops on the Internet ☆

Jianhong Xia, Lixin Gao *, Teng Fei

Department of Electrical and Computer Engineering, University of Massachusetts at Amherst, Amherst, MA 01003, United States

Received 18 January 2007; received in revised form 4 July 2007; accepted 21 July 2007
Available online 3 August 2007

Responsible Editor: D. Medhi

Abstract

In this paper, we present a measurement study of persistent forwarding loops and a flooding attack that exploits persistent forwarding loops. Persistent forwarding loops may share one or more links with forwarding paths to some hosts. An attacker can exploit persistent forwarding loops to overload the shared links and disrupt Internet connectivity to those hosts.

To understand the extent of this vulnerability, we perform extensive measurements to systematically study persistent forwarding loops. We find that persistent forwarding loops do exist in the Internet. At least 35 million addresses experience persistent forwarding loops, and at least 11 million addresses can be attacked by exploiting such persistent forwarding loops. In addition, 87.4% of persistent forwarding loops involve routers in destination domains, which can be observed from various locations. This makes it possible to launch attacks from multiple vantage points. We also find that most persistent forwarding loops are just two hops long, which enables an attacker to significantly amplify traffic to them.

We further investigate the possible cause of persistent forwarding loops, and find that about 50% of them are caused by neglecting to configure pull-up routes. We show that even if the misconfiguration occurs in a stub network, it may cause persistent forwarding loops involving routers in large ISPs, and can potentially be exploited by attackers to flood links in a backbone network. To the best of our knowledge, this is the first study of exploiting routing misconfigurations to launch DDoS attacks and understanding the impact of such attacks.

Keywords: Internet routing; Persistent forwarding loop; Flooding attacks; Distributed denial of service; Misconfiguration

☆ Part of this work has been published in the Internet Measurement Conference in 2005 (IMC'05) [13] as a short paper and presented in the North American Network Operators' Group (NANOG36) [14]. This work was supported in part by NSF Grants CNS-0325868, CNS-0208116 and Alfred P. Sloan Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of National Science Foundation or Alfred P. Sloan Fellowship.
* Corresponding author. Tel.: +1 413 545 4548; fax: +1 413 545 1993.
E-mail addresses: jxia@ecs.umass.edu (J. Xia), lgao@ecs.umass.edu (L. Gao), tfei@ecs.umass.edu (T. Fei).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2007.07.004

1. Introduction

One of the most prevalent threats in the Internet is the Distributed Denial of Service (DDoS) attack. In general, DDoS attacks send traffic from a large number of compromised hosts to deplete the victim's network or host resources. In this paper, we present flooding attacks that exploit persistent forwarding loops in the Internet. Forwarding loops have been observed in previous measurement studies [5,9,11,15]. Although transient forwarding loops will disappear after routing protocols converge, forwarding loops caused by configuration errors can last for a long time. In addition to the obvious issues that persistent forwarding loops can blackhole network addresses, they also can be exploited to overload the links in which they are involved. Since persistent forwarding loops may share one or more links with forwarding paths to some hosts, overloading those shared links can disrupt Internet connectivity to those hosts, which may lead to denial of service attacks.

Fig. 1 shows an example of a flooding attack that exploits persistent forwarding loops. Traffic to host X traverses routers R_a, R_b, R_c and other network devices to reach the destination. At the same time, traffic to host Y also traverses routers R_a, R_b and R_c. However, due to misconfigurations in router R_c, traffic to host Y in router R_c will be forwarded back to R_b. Therefore, any packet destined to address Y falls into the loop between R_b and R_c, and will be dropped only when its time-to-live (TTL) expires. In this scenario, the traffic volume to host Y will be amplified in link L_{bc}, so that link L_{bc} can be overloaded if malicious attackers deliberately send a large amount of traffic to host Y. Therefore, host X would experience denial of service. Since traffic traversing a persistent forwarding loop typically traverses the links in the loop multiple times before being dropped, it takes attackers much less effort to launch flooding attacks, making the attacks stealthy. Since network operators can see traffic congestion only on the shared link L_{bc} but not on other links such as L_{ab}, without packet-level or flow-level measurements on the shared link, this kind of attack is hard to detect.

To understand the extent of this vulnerability, we perform extensive measurements to systematically study persistent forwarding loops. We find that persistent forwarding loops do exist in the Internet. We characterize the prevalence of IP addresses involving persistent forwarding loops and the properties of such loops in terms of length and location. The major contributions of our paper are summarized as follows,

1. *Prevalence of IP addresses involving persistent forwarding loops*: In our measurement, we find that at least 35 million addresses, i.e., 2.47% of routable addresses, experience persistent forwarding loops. At least 11 million addresses, i.e., 0.78% of routable addresses, can be attacked by exploiting persistent forwarding loops. These addresses are spread widely over many domains.
2. *Properties of persistent forwarding loops*: Our results show that about 94.3% of persistent forwarding loops happen within a single domain, while about 5.7% happen across multiple domains. In addition, about 87.4% of persistent forwarding loops involve routers in destination domains and they can be observed from various locations. This makes it possible to launch attacks from multiple vantage points. We also find that about 90% of persistent forwarding loops are just two hops long, which enables an attacker to significantly amplify traffic to persistent forwarding loops.
3. *Possible cause of persistent forwarding loops*: We investigate the possible cause of persistent forwarding loops, and find that about 50% are caused by neglecting to configure pull-up routes.

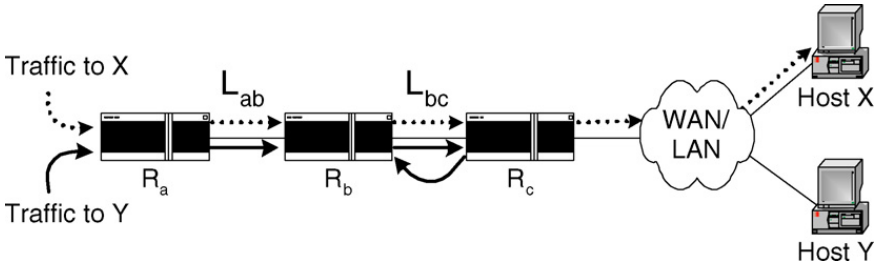


Fig. 1. Flooding attacks by exploiting persistent forwarding loops.

4. *Impact of persistent forwarding loops on Tier-1 ASes*: We find that all Tier-1 autonomous systems (ASes) have one or more routers involved in persistent forwarding loops. This enables attackers to exploit the persistent forwarding loops and interfere with backbone networks in Tier-1 ASes. We show that even if a misconfiguration occurs in a stub network, it may cause persistent forwarding loops involving routers in large ISPs, and can potentially be exploited by attackers to flood links in a backbone network.

The remainder of this paper is structured as follows. Section 2 introduces the terminology of persistent forwarding loops, network addresses that have persistent forwarding loops and network addresses that can be attacked by persistent forwarding loops. We describe the measurement design in Section 3, and characterize the properties of persistent forwarding loops in terms of length and location in Section 4. Section 5 discusses the possible cause of persistent forwarding loops and identifies the loops that are caused by neglecting to configure pull-up routes. We conclude the paper and discuss our future work in Section 6.

2. Terminology

In this section, we define some terminologies that are used in this paper. First we explain the concepts of forwarding loop and persistent forwarding loop, then we define shadowed address and imperiled address followed by an example from the real network.

2.1. Forwarding loops

In general, a packet from source s traverses a sequence of routers to reach destination d. A packet experiences a *forwarding loop* if it traverses a set of routers more than once. We denote the sequence of router interfaces that a packet from s to d traverses as $(r_1, r_2, \ldots, r_n)$, which is also called *forwarding path* from s to d. If $r_i = r_j$ and $i \neq j$, then the forwarding path contains a forwarding loop $(r_i, \ldots, r_j)$ from s to d. The *length of forwarding loop* $(r_i, \ldots, r_j)$ from s to d is $j - i$.

Forwarding loops can be transient or persistent. *Transient forwarding loops* are those that can resolve themselves without human intervention or network topology changes. Transient forwarding loops may occur during routing protocol convergence [2]. Hengartner et al. [5] have demonstrated that forwarding loops exist in the Sprint backbone network by analyzing packet traces. In general, transient forwarding loops will disappear after the routing protocols converge. However, some forwarding loops will not disappear without human intervention or network topology changes. We refer to those forwarding loops as *persistent forwarding loops*.

2.2. Shadowed addresses and imperiled addresses

If there is a persistent forwarding loop from source s to destination d, we refer to the IP address of d as a *shadowed address*. For example, the address Y in Fig. 1 is a shadowed address. Note that the links in persistent forwarding loops may still be able to carry traffic to some reachable hosts. That is, a persistent forwarding loop may share one or more links with forwarding paths to addresses other than the shadowed addresses. If a destination d' is reachable and the forwarding path to d' shares a link with a persistent forwarding loop, we refer to the address of d' as an *imperiled address*. For example, the address X in Fig. 1 is an imperiled address. Imperiled addresses are named so because they may suffer from potential threats posed by the persistent forwarding loops. In other words, if flooding attacks are launched by sending traffic to shadowed addresses, the links in persistent forwarding loops will be overloaded. The imperiled addresses will either experience performance degradation in network connections or even be disconnected.

Examples of a shadowed address and an imperiled address are shown in Table 1. In this example,

Table 1
An example of shadowed address and imperiled address

Hop	Traceroute to shadowed address 81.181.31.127	Traceroute to imperiled address 80.96.192.10
1	128.119.91.254	128.119.91.254
2	128.119.2.238	128.119.2.238
3	128.119.2.194	128.119.2.194
4	65.77.95.161	65.77.95.161
...	...	...
18	166.49.147.134	166.49.147.134
19	195.39.208.82	195.39.208.66
20	193.226.179.18	193.226.179.18
21	**193.226.130.226**	**193.226.130.226**
22	**194.176.189.42**	**194.176.189.42**
23	**193.226.130.226**	194.105.11.178
24	**194.176.189.42**	80.96.192.10
25	**193.226.130.226**	
26	**194.176.189.42**	
...	...	

from a host in the campus network, 128.119.0.0/16, in the University of Massachusetts at Amherst (UMass), traffic to the shadowed address 81.181.31.127 falls into a persistent forwarding loop between routers 193.226.130.226 and 194.176.189.42, while traffic to the imperiled address 80.96.192.10 relies on that link to reach the destination.

3. Data collection

3.1. Measurement design

We use traceroute to discover forwarding paths in our study. Our goal is to identify all possible IP addresses in the Internet that have persistent forwarding loops. In order to reduce the measurement overhead, we select a set of representative network addresses to do traceroute. We select as few network addresses as we could, while trying to explore as many forwarding paths as possible. Most networks are allocated by a set of contiguous IP addresses. Since it is common that the forwarding decision in a router is based on the destination IP address, tracing different IP addresses in a network from the same source may observe the same forwarding path. In order to discover forwarding paths to a network, we could sample just a few addresses in that network. However, the address space of a large network may be aggregated by multiple address blocks from its customers. Sampling a limited number of addresses in such large networks might not discover all forwarding paths to those customers. On the other hand, even if the address space of a large network is not aggregated from its customers, when the network owns a large number of sub-networks, forwarding paths to various subnets may still be different. Hence, sampling a limited number of addresses in a network may not be able to discover as many distinct forwarding paths as possible. We note that it is quite common for a large network to use /24 subnet as the smallest unit to divide its network into multiple subnets. In addition, it is reasonable to assume that forwarding paths to the addresses within a subnet should be the same. We believe that sampling addresses in any /24 address block is efficient and fine-grained enough to discover as many distinct forwarding paths to a large network as possible. Therefore, we design our measurement by selecting a few IP addresses in each address block no larger than /24 to perform traceroute.

It should be noted that not all IP addresses have been allocated. Even if we can obtain a list of all allocated IP addresses from the Internet Assigned Numbers Authority (IANA) [6], not all allocated IP addresses have been used in the Internet. In general, we can get all addresses that have been used in the Internet from a router in default-free zone. We refer to the prefixes in BGP routing tables in default-free zone as *routable prefixes*, and refer to the addresses covered by routable prefixes as *routable addresses*. In our measurement study, we use BGP routing tables in the RouteViews Project [12] to get routable prefixes and routable addresses, because it peers with BGP routers in many large ISPs such as AT&T and Sprint.

We partition all routable addresses into blocks which contain at most 256 addresses. In other words, we divide all routable prefixes whose length is shorter than 24 into multiple /24 prefixes. For example, prefix 12.0.0.0/8 is divided into 65,536 prefixes represented by 12.x.x.0/24. All prefixes in our measurement have a length of at least 24. We refer to these prefixes as *fine-grained prefixes*.

Since the forwarding paths to only a limited number of sampled IP addresses are used to represent forwarding paths to all IP addresses in a fine-grained prefix, we extend the concept of persistent forwarding loops to fine-grained prefixes. If an address d_p in a fine-grained prefix p experiences persistent forwarding loops from source s, we say there is a persistent forwarding loop from s to prefix p, and refer to prefix p as a *shadowed prefix*. Similarly, if a fine-grained prefix q contains an imperiled address, we refer to prefix q as an *imperiled prefix*.

3.2. Data sets

In September 2005, we collected two sets of trace data from one location. We also collected additional traces from various locations to support our findings. Unless otherwise specified, all traces were collected from a subscriber of Comcast high speed Internet service in western Massachusetts.

The first data set, D_A, identifies all fine-grained prefixes to which there is a forwarding loop. From a total of 0.18 million routable prefixes, we obtained about 5.51 million fine-grained prefixes. Due to security and privacy concerns posed by networks owned by governmental and military agencies, we filtered out their prefixes according to WHOIS [1]. After filtering, 5.5 million fine-grained prefixes were

Table 2
Summary on measurement design and trace data

Data set	Fine-grained prefix selection	# of selected IP addresses per prefix	# of traces for each selected address	# of prefixes traced	# of addresses traced
D_A	All fine-grained prefixes	2	Once	5,499,618	10,999,236
D_B	All candidate prefixes	2	10 times	207,891	415,782

traced. To reduce the overhead of our measurement, we performed traceroute to two IP addresses in each prefix, the first one and a random one. We collected about 11 million traces in D_A within 16 days. We refer to those fine-grained prefixes with forwarding loops in D_A as *candidate prefixes*.

The second data set, D_B, identifies all shadowed prefixes to which there is a persistent forwarding loop. To identify persistent forwarding loops, we traced to candidate prefixes and performed traceroute multiple times. Although we could observe forwarding loops from a single trace, it was impractical to monitor the network forever to identify persistent forwarding loops. Thus, we adopted an approximate criterion with respect to the general time scale of routing convergence. We traced an IP address d multiple times within 6 days. If the forwarding loop always appeared in all traces to d, we classified the forwarding loop as a persistent forwarding loop. To collect D_B, we traced to the candidate prefixes and selected two IP addresses in each prefix, a first one and a random one. Each selected IP address was traced 10 times. We collected about 4.2 million traces by tracing to 415,782 IP addresses. Note that we did not see much difference in the observations from the first day to the last day in this experiment. It means that there was no noticeable change on the forwarding paths we collected during the 6-day measurement period. Table 2 summarizes the data sets of D_A and D_B.

4. Characterizing persistent forwarding loops

A trace of traceroute normally contains a sequence of router interface addresses. However, some traces may contain "*" or "!" when routers do not send back ICMP packets, replies get lost or filtered, or destinations cannot be reached. To reduce ambiguity, we filtered out the traces that contain "*" or "!" between two appearances of the same address. We found that a few traces contained the same address appearing continuously, which could be caused by a firewall. We also filtered out these traces in our study.

4.1. Prevalence of shadowed prefixes and imperiled prefixes

To understand the scope of persistent forwarding loops in the Internet, we look at the prevalence of shadowed prefixes and imperiled prefixes from our experiments. We also present the distributions of shadowed prefixes and imperiled prefixes to show how these network addresses spread in the IPv4 address space.

4.1.1. Shadowed prefixes

In our measurement, we identified the candidate prefixes from D_A and performed traceroute to these candidate prefixes to collect D_B. We then analyzed D_B to identify shadowed prefixes and persistent forwarding loops.

Among 5.5 million prefixes traced in D_A, 207,891 of them were identified as candidate prefixes, which cover about 3.77% of routable IP addresses. From data D_B that traces to all candidate prefixes, we identified 135,973 prefixes as shadowed prefixes, which cover about 2.47% of routable IP addresses, i.e., about 35 million addresses. Shadowed prefixes are located in 5341 ASes, which suggests that IP addresses experiencing forwarding loops originate from a large number of ASes.

4.1.2. Imperiled prefixes

As mentioned in Section 2, the vulnerability of persistent forwarding loops does not come from the shadowed addresses themselves. Rather, it comes from the shared links between persistent forwarding loops and the forwarding paths to imperiled addresses. To understand the extent of this vulnerability, we estimated the prevalence of imperiled addresses in the Internet.

The basic idea of identifying imperiled addresses is to find those IP addresses that are reachable, and whose forwarding paths share one or more links with persistent forwarding loops. It is not easy to fully identify the imperiled addresses in the Internet without a global view of forwarding paths from a source to a destination. In our experiment, we

estimated the number of imperiled addresses/prefixes from D_A. Any reachable address in D_A that uses one or more links in a persistent forwarding loop is marked as an imperiled address. The fine-grained prefixes containing any imperiled address are marked as imperiled prefixes. Based on the persistent forwarding loops found in Section 4.1.1 and the traces in D_A, 42,887 of fine-grained prefixes were identified as imperiled prefixes. It covers about 0.78% of routable IP addresses, i.e., about 11 million addresses. These imperiled addresses could be potential victims when the vulnerability on persistent forwarding loops is exploited. These imperiled prefixes originate from 2117 ASes, so the potential victims are spread widely in various domains.

Note that not all persistent forwarding loops share their links with forwarding paths to imperiled prefixes. Among 302,989 persistent forwarding loops, only 24.1% of them share links with forwarding paths to imperiled addresses. We call those shadowed addresses (prefixes) that can be exploited to attack imperiled addresses *dark addresses* (*prefixes*). Among 135,973 shadowed prefixes, 18.4% of them are dark prefixes. Generally, a persistent forwarding loop shares links with forwarding paths to one or two imperiled prefixes. However, some persistent forwarding loops may share links with forwarding paths to as many as 1000 imperiled prefixes. Flooding such shared links can result in denial of service to a large number of imperiled addresses.

4.1.3. Distribution of shadowed addresses and imperiled addresses

To understand which addresses experience persistent forwarding loops and which addresses can be attacked by exploiting persistent forwarding loops, we plot the number of shadowed addresses and the number of imperiled addresses based on the distribution of the first byte of their addresses. Fig. 2 plots the number of routable addresses, the number of shadowed addresses and the number of imperiled addresses on a log scale.

In Fig. 2, we can see that routable addresses are not uniformly distributed within the entire IPv4 address space. Some address blocks are reserved for use by multicast applications, while some have not been allocated by IANA yet [6]. Similar to the routable addresses, shadowed addresses and imperiled addresses are not uniformly distributed within the entire IPv4 address space either. However, they are roughly proportional to the distribution of routable addresses. Shadowed addresses and imperiled addresses almost fall into each /8 prefix that has been allocated by IANA. It confirms our observations in Sections 4.1.1 and 4.1.2. That is, the shadowed addresses and imperiled addresses are widely distributed in the various networks. One interesting observation is that although we do not see any shadowed addresses in 20.0.0.0/8, we do observe many imperiled addresses in 20.0.0.0/8. This means that some addresses can be attacked even when the addresses close to them do not experience persistent forwarding loops. We also find that although some prefixes contain shadowed addresses, there are no imperiled addresses in them. For example, prefix 56.0.0.0/8 contains many shadowed addresses, but there are no imperiled addresses in it.

4.2. Observing persistent forwarding loops to additional addresses in shadowed prefixes

As mentioned in Section 3, we chose two IP addresses in each fine-grained prefix to discover the forwarding paths to the entire IP addresses in that prefix. To demonstrate that persistent forwarding loops exist to other addresses in shadowed prefixes, we traced to 3705 shadowed prefixes from the campus network in the University of Massachusetts at Amherst, and selected about 50 random IP

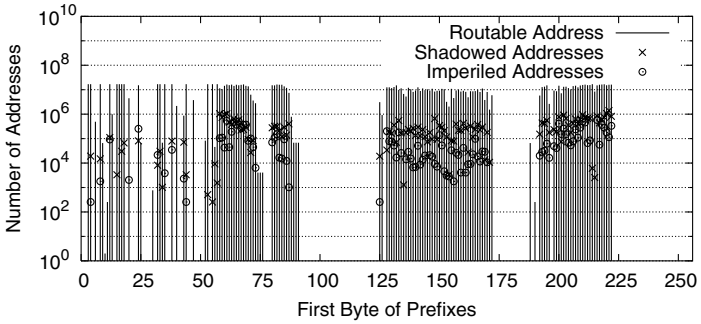


Fig. 2. Distribution of routable addresses, shadowed addresses and imperiled addresses.

Table 3
Persistent forwarding loops to additional sampled addresses

Category	# of shadowed prefixes	Percentage (%)
All sampled addresses have forwarding loops	**2401**	**68.96**
Partial sampled addresses have forwarding loops	**1083**	**30.65**
(a) Infrastructure addresses are sampled	791	–
(b) Unknown reason	292	–
None of sampled addresses has forwarding loops	**49**	**1.39**
Total	**3533**	**100**

addresses in each prefix instead of two addresses. We denote this trace as D_C. As mentioned before, we filtered out the traces that contained "*" or "!" between two appearances of a same address, and finally we have traces to 3533 shadowed prefixes.

The result shows that most additional sampled addresses also experience forwarding loops. 67.96% of shadowed prefixes in D_C confirm that all additional sampled addresses have forwarding loops. We further investigated the reason why not all additional sampled addresses had forwarding loops in shadowed prefixes. We found that 73.41% of them are caused by the fact that infrastructure addresses (deployed for router interfaces) are sampled. For example in Fig. 1, although there is a forwarding loop when tracing to host Y, there is no forwarding loop if we trace to the interface address of R_c. The result is shown in Table 3. This suggests that the persistent forwarding loop is not a special phenomenon of the sampled addresses, but instead, it can be applied to most addresses in shadowed prefixes.

4.3. Observing persistent forwarding loops from multiple vantage points

To show that persistent forwarding loops can be observed from multiple locations, we traced to 4894 shadowed prefixes from various vantage points, and selected four random IP addresses in each prefix. We collected the traces from four hosts in PlanetLab [10] that are located in Asia, Europe, and the east and west coasts of the United States. We denote data sets from these four hosts as D_{D1}, D_{D2}, D_{D3}, and D_{D4}, respectively. We found that, persistent forwarding loops to about 90% of shadowed prefixes can still be observed from all four locations, which are shown in Table 4. This suggests that persistent forwarding loops can be observed not only from the location of our measurement point, but also from multiple vantage points.

Table 4
Persistent forwarding loops from different locations

Data set	Location	# of prefixes traced	# of prefixes with loops	Percentage (%)
D_{D1}	Asia	4894	4315	88.10
D_{D2}	Europe	4894	4262	87.02
D_{D3}	US East Coast	4894	4543	92.83
D_{D4}	US West Coast	4894	4516	92.20

If attackers exploit persistent forwarding loops from different locations, and choose distinct IP addresses in shadowed prefixes to overload the links in the loops, such flooding attacks would be difficult to detect because the traffic would be coming from different sources and toward different destinations. It makes this vulnerability even more critical.

4.4. Length of persistent forwarding loops

In order to understand the traffic amplification factor in the links that appear in the persistent forwarding loops, we must first understand their length. When a packet enters a persistent forwarding loop, it may traverse the links in the loop multiple times before its TTL expires. The shorter the length of a persistent forwarding loop, the more times a packet traverses the links in the loop. We found that, among 302,989 persistent forwarding loops, over 89.4% of them had a length of 2, which could significantly amplify the traffic to the shadowed addresses in the links that appeared in the persistent forwarding loops. About 10.4% of them had a length from 3 to 9. The rest had a length of 10 or longer. Several persistent forwarding loops had a length as long as 16. The distribution of length of persistent forwarding loops is shown in Fig. 3.

4.5. Location of persistent forwarding loops

Identifying the location of persistent forwarding loops is helpful for us to understand where they occur. Persistent forwarding loops may occur within the destination domains, or across one or more other domains. To understand where these persistent forwarding loops are located and how many domains are involved, we classify the persistent

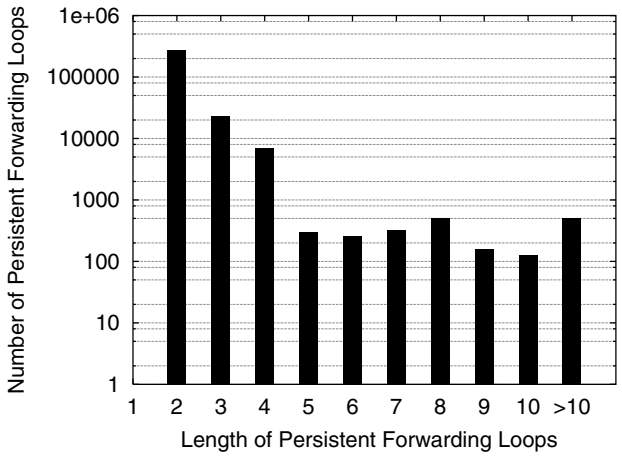


Fig. 3. Distribution of length of persistent forwarding loops.

forwarding loops based on two criteria. The first is whether the persistent forwarding loops involve routers in destination domains. The other is whether persistent forwarding loops involve routers across multiple domains.

It is difficult to accurately map infrastructure IP addresses to AS numbers [8]. The inaccuracies may occur at AS boundaries. We use several heuristic algorithms to justify our results in classifying persistent forwarding loops.

4.5.1. Whether persistent forwarding loops involve routers in destination domains

We consider that a persistent forwarding loop involves router(s) in the destination domain if a router interface address exists in the persistent forwarding loop and originates from the same domain as the shadowed address. For example, suppose that a forwarding path to the destination d is $(r_1, r_2, \ldots, r_n)$, which contains a persistent forwarding loop $(r_i, \ldots, r_j)$. If at least one of the router interface addresses in $(r_i, \ldots, r_j)$ originates from the same domain of d, we say that the persistent forwarding loop involves routers in the destination domain. Otherwise, we consider that the destination domain is not involved in the persistent forwarding loop.

Based on this classification, we found that about 87.44% of persistent forwarding loops involve one or more routers in the destination domains, and 12.56% of persistent forwarding loops do not involve any router in the destination domains.

We further investigated the persistent forwarding loops that involve routers in the destination domains. We found that about 3.78% of persistent forwarding loops contain an interface address of one router only in the destination domains, and 83.66% of persistent forwarding loops contain interface addresses of two or more routers in the destination domains. As we mentioned before, inaccuracy of mapping infrastructure IP addresses to AS numbers may occur at AS boundaries. It is possible to map one non-destination domain border router into destination domain, but if we have interface addresses of two or more routers in the loop mapped into the destination domains, the chance of incorrectly mapping both routers is quite small. Therefore, the classification error on persistent forwarding loops that involve routers in the destination domains should be less than 3.78%.

Among persistent forwarding loops that contain interface addresses of two or more routers in the destination domains, we look at a special case, interface addresses of all routers in the loop $(r_i, \ldots, r_j)$ originate from the same domain as the destination d. That is, all involved routers in the loop originate from the destination domain. Given a forwarding path in a trace, $(r_1, r_2, \ldots, r_n)$, if it contains a persistent forwarding loop $(r_i, \ldots, r_j)$, we refer to the router interface r_{i-1} as the *preceding router interface* of the loop in this trace. To reduce the error of mapping IP addresses to AS numbers, we take account of the preceding router interface of the loop in the traces. If the preceding router interface r_{i-1} and all addresses in the loop $(r_i, \ldots, r_j)$ originate from the same domain as d, then we say the persistent forwarding loops are within destination domain. Using this heuristic, we found that about 58.21% of persistent forwarding loops occurred within the destination domains.

For persistent forwarding loops that do not involve any router in the destination domains, we verify the domain of its preceding router interface r_{i-1} in the traces. If address r_{i-1} originates from the destination domain, it is possible that router r_i is located in the destination domain, but we map it into a non-destination domain. However, if address r_{i-1} is not originated from the destination domain, the chance of address r_i being in the destination domain is quite small. The classification error of this category could happen only when the preceding router interface of the loop is in the destination domain. We found that only 1.47% of persistent forwarding loops do not involve any routers in the destination domain but the preceding router interface of the loop in the trace is originated from the destination domain. Therefore, the classification

Table 5
Classifying persistent forwarding loops based on whether destination domain is involved

Category of persistent forwarding loops	Percentage (%)
1. Destination domain is involved	**87.44**
i. loops contains only one address in the destination domain	3.78
ii. loops contain two or more addresses in the destination domain	83.66
— All addresses in the loop are in the destination domain	*58.21*
2. Destination domain is not involved	**12.56**
The preceding router interface of the loop in the traces	
i. is in the destination domain	1.47
ii. is not in the destination domain	11.09
Total	**100**

Table 6
Classifying persistent forwarding loops based on the number of involved domains

Category of persistent forwarding loops	Percentage (%)
1. Only a single domain is involved	**94.27**
The preceding router interface of the loop in the traces	
i. is in the same domain as the loop	67.06
ii. is not in the same domain as the loop	27.21
2. Multiple domains are involved	**5.73**
i. Two domains are involved	5.35
ii. Three or more domains are involved	0.38
Total	**100%**

error on the persistent forwarding loops that do not involve routers in the destination domains is at most 1.47%. The result is shown in Table 5.

4.5.2. Whether persistent forwarding loops involve multiple domains

To understand the scope of persistent forwarding loops, we classify the persistent forwarding loops based on the number of involved domains. For example, suppose that a forwarding path $(r_1, r_2, \ldots, r_n)$ contains a persistent forwarding loop $(r_i, \ldots, r_j)$, we calculate the number of distinct domains that routers $(r_i, \ldots, r_j)$ originate from. If all addresses of $(r_i, \ldots, r_j)$ have the same AS number, we consider the persistent forwarding loop involves in a single domain.

We found that about 94.27% of persistent forwarding loops occur within a single domain, and 5.73% of persistent forwarding loops occur among multiple domains. Similar to the previous subsection, we investigate the preceding router interface r_{i-1} of the loop in the traces and estimate the classification error that might happen at AS boundaries. If the preceding router interface r_{i-1} originates from the same domain as all routers in the loop, the chance of misclassification on mapping address r_i is quite small. We found that 67.06% of persistent forwarding loops occur in a single domain with the preceding router interface r_{i-1} in the same domain. About 27.21% of persistent forwarding loops occur in a single domain with the preceding router interface r_{i-1} not in the same domain as the loop.

For the persistent forwarding loops that involve multiple domains, we found that 5.35% of them involve only two domains. Only 0.38% involve three or more domains. The result is shown in Table 6.

4.6. Impact on Tier-1 ASes

Although the number of persistent forwarding loops that occur in multiple domains is quite small, they can impact the inter-domain links among involved domains. Most persistent forwarding loops that occur in multiple domains contain one or more routers in the large ISPs. If the links in persistent forwarding loops are overloaded by flooding attacks, it may have a serious impact on large ISPs or links in a backbone network.

To understand the risk of persistent forwarding loops, we investigated how many large ISPs and/or Tier-1 ASes could be impacted. We used the heuristic algorithm in [3] to identify 19 Tier-1 ASes, and found that all these Tier-1 ASes have one or more routers involved in persistent forwarding loops.

We also used DNS to resolve the router interfaces in the persistent forwarding loops to obtain the domain name. Among 82,626 router interface addresses, 52.4% of them were successfully resolved by DNS, confirming that all Tier-1 ASes, such as AS701 (UUnet Technologies, Inc), AS7018 (AT&T WorldNet Services), AS1239 (Sprint) and AS3356 (Level 3 Communications, LLC), have routers involved in persistent forwarding loops. Most large ISPs, including Qwest, Verio, SBC Global, Savvis and GLBX, also have routers involved in the loops.

Based on the domain name of resolved routers, we found that these routers are widely distributed in 129 countries, including the US, Japan, Brazil,

Russia, Germany, Italy and Mexico. The impact of persistent forwarding loops is not limited to one domain, or one country or one network. It may impact the Internet everywhere in terms of geographical location and ISPs.

4.7. Flooding attacks using persistent forwarding loops

In this subsection, we analyze the impact on the bandwidth consumption of the links in persistent forwarding loops and the effort that would be required of an attacker in order to launch such flooding attacks.

When a packet is sent to a shadowed address, it will fall into the persistent forwarding loop and will be dropped only when its TTL expires. Therefore the packet may traverse the links in the loop multiple times before being dropped and will consume more bandwidth. We define *traffic amplification factor* as the average number of times that a packet traverses a link in a persistent forwarding loop. Typically, a malicious packet could have a TTL value of 255 when created at its origin. From our measurement, we find that persistent forwarding loops occur on average 14 or 15 hops away from the source. Without losing generality, we consider that a packet traverses about 14 routers to fall into persistent forwarding loops. The persistent forwarding loops typically have a length of 2 as shown in Section 4.4. With this statistic, the traffic amplification factor can be estimated to be $\frac{255-14}{2} = 120$. This means that a packet will traverse the links in forwarding loops 120 times that which is expected. Even for a long loop with 16 hops, the traffic amplification factor can also be 15 times. Thus persistent forwarding loops can create much more traffic than might be expected.

Due to the amplification of traffic by persistent forwarding loops, attackers would require much less effort to launch flooding attacks on imperiled addresses. For example, in Fig. 1, if the available bandwidth for the link L_{bc} is 50Mbps, and traffic amplification factor is 25, then an attacker needs to send traffic at the rate of 2Mbps to flood L_{bc}. If an attacker has compromised 100 computers in the Internet and launches such an attack, the average traffic rate on each machine is only 20Kbps. Such a rate can be easily handled by most users and would be hard to detect from the source.

5. A possible cause of persistent forwarding loops

It is hard to identify the root causes of persistent forwarding loops without information about configurations on the involved routers. We conjecture that the most possible cause of persistent forwarding loops is misconfiguration of the common usages on default routes and static routes. Several examples in [4] have shown that forwarding loops can happen if BGP or static routes are incorrectly configured. BGP misconfigurations are common today in the Internet [7].

5.1. Neglecting to configure pull-up routes

To understand how misconfigurations can easily lead to persistent forwarding loops, we show an example in which a network administrator neglects to configure a "pull-up route" at a border router to his upstream provider. In Fig. 4, provider P owns 18.0.0.0/8 and delegates 18.1.0.0/16 to its customer *C*. The provider's border router has a static route directing traffic for 18.1.0.0/16 to the customer's border router. The customer's border router, in turn, has routes for some subnets of 18.1.0.0/16, such as 18.1.1.0/24 and 18.1.2.0/24, but not for others. The customer's border router also configures a default route (e.g., 0.0.0.0/0) pointing to the link back to the provider's router, for access to the Internet. That would cause a persistent forwarding loop for all traffic destined to addresses in the range from 18.1.3.0 to 18.1.255.255.

In the above case, the forwarding loop is two hops long and near the destination domain. However, when the customer C is multi-homed, the same misconfiguration may also lead to a persistent forwarding loop that occurs across multiple domains. For example, the customer C has another provider B and prefers to use its link to provider B for outbound traffic. Therefore, the customer C's border router has a default route 0.0.0.0/0 to provider B. The customer C also prefers to use the link from provider P for inbound traffic. In this case, when the customer C receives any traffic destined to addresses in the range from 18.1.3.0 to 18.1.255.255, it will forward the traffic to its provider B. The provider B will forward the traffic to its own provider or neighbors. Eventually the traffic will return to the provider P and reach the customer C again. That would cause a persistent forwarding loop occurring across

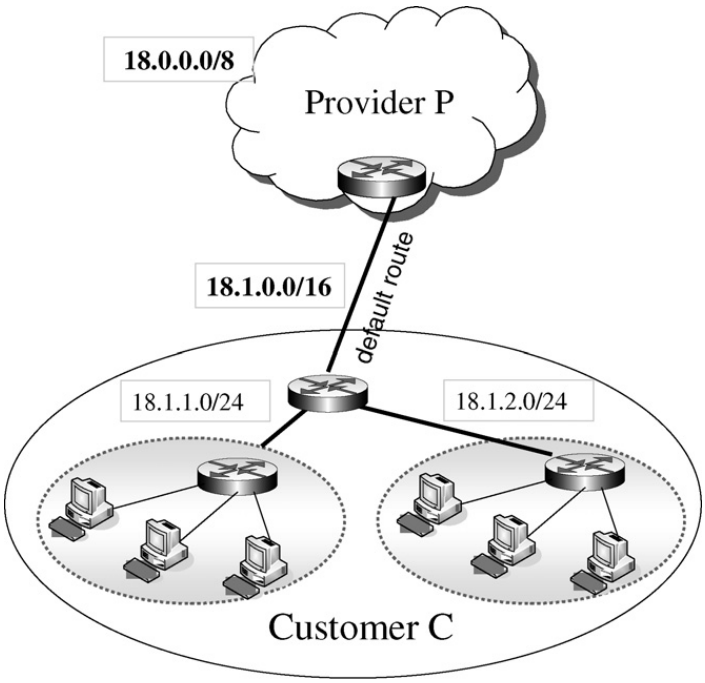


Fig. 4. Example on neglecting to configure pull-up routes.

multiple domains which may have a loop length much longer than 2.

To prevent this, the customer needs to configure a "null route" for 18.1.0.0/16 to discard packets to any destinations in 18.1.0.0/16 that do not have a more specific route.

5.2. Case studies

We show several cases in this subsection that demonstrate the persistent forwarding loops that occur within one domain, between two domains, and across three or more domains.

5.2.1. Persistent forwarding loops within one domain

First, we look at an example of a persistent forwarding loop that occurs within one domain. The example shown in Fig. 5 illustrates the persistent forwarding loop within the destination domain. Network addresses 69.33.0.0/16 are owned by MegaPath Networks Inc (AS 23215), and are announced to the Internet. However, when we tra-

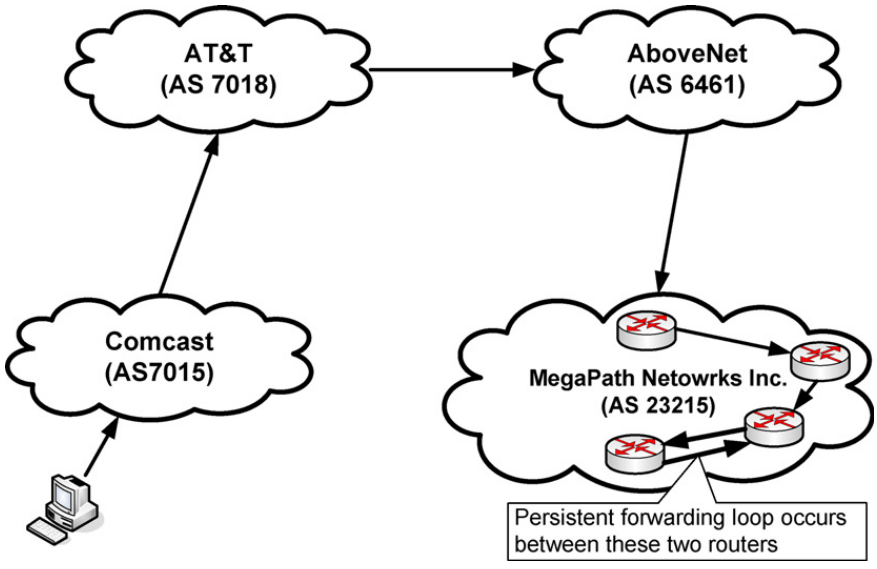


Fig. 5. Persistent forwarding loop within a single domain.

Table 7
An example of traces that contain persistent forwarding loops within a single domain

Hop	Router address	Router name
...	...	...
11	12.122.10.106	tbr2-cl5.cgcil.ip.att.net
12	12.123.216.217	gar1-p390.chail.ip.att.net
13	12.119.137.50	n/a
14	64.125.30.146	so-1-0-0.cr1.ord2.us.above.net
15	64.124.229.53	ge-0-1-0.core1.chi.megapath.net
16	66.80.132.142	ve308.ge1-2-0.core1.dfw.megapath.net
17	66.80.128.106	fe-1-4.edge1.dfw.megapath.net
18	**66.80.128.101**	**fe-1-3-0.core1.dfw.megapath.net**
19	66.80.128.106	fe-1-4.edge1.dfw.megapath.net
20	**66.80.128.101**	**fe-1-3-0.core1.dfw.megapath.net**
21	66.80.128.58	fe1-1.edge1.dfw.megapath.net
22	**66.80.128.101**	**fe-1-3-0.core1.dfw.megapath.net**
23	66.80.128.102	fe-1-3.edge1.dfw.megapath.net
...	...	...

ceroute to 69.33.53.1 from an end-user of Comcast Network, we find there is a persistent forwarding loop inside MegaPath Networks. Table 7 shows the result of the traceroute. Since some addresses in 69.33.0.0/16 are reachable in MegaPath Networks, and some addresses in 69.33.0.0/16 have persistent forwarding loops within MegaPath Networks, the configured router should be within the destination domain.

5.2.2. Persistent forwarding loops between two domains

In this subsection, we look at an example of a persistent forwarding loop that occurs between two domains. The example shown in Fig. 6 illustrates the case. The network addresses 216.228.0.0/19 are owned by Red Shift, Inc. (AS 7735). Red Shift announces these addresses using 32 /24 prefixes. However, when we traceroute to 216.228.25.1, there is a forwarding loop between two routers, one is 216.228.2.73 (cs0-FE-0-0.corp. redshift.net) and the other one is 216.171.145.66 (cust-router-redshift.transedge.com). The latter router belongs to Red Shift's provider, New Edge Networks (AS 19029). Table 8 shows the result of the traceroute, which suggests that the forwarding loop occurs at the border routers between Red Shift and its provider New Edge Networks.

5.2.3. Persistent forwarding loops across three or more domains

In this case, we look at an example of a persistent forwarding loop that occurs across four domains. The example is shown in Fig. 7. The network addresses 72.9.192.0/22 is owned by ClearSky Broadband (AS 33269). ClearSky Broadband announces 72.9.192.0/22 to its provider Williams Communications, Incorporated (AS 7911). However, ClearSky Broadband uses only half space, 72.9.192.0/24 and 72.9.193.0/24, in its own network, and assigns another half space, 72.9.194.0/24 and 72.9.195.0/24, to its customer, Videsh Sanchar Nigam Ltd (VSNL) in India (AS 4755). We find that VSNL uses only 72.9.195.0/24 address space, and does not use 72.9.194.0/24 address space at all. In addition, VSNL does not drop any traffic destined

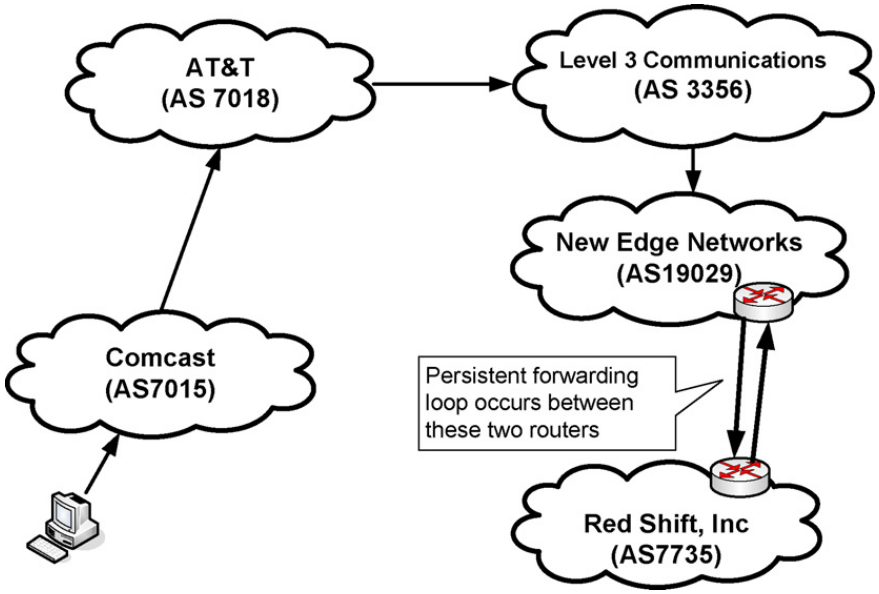


Fig. 6. Persistent forwarding loop between two domains.

Table 8
An example of traces that contain persistent forwarding loops between two domains

Hop	Router address	Router name
...	...	...
11	12.122.10.22	tbr2-cl16.n54ny.ip.att.net
12	12.123.3.109	ggr2-p3120.n54ny.ip.att.net
13	4.68.127.149	so-8-1.car3.NewYork1.Level3.net
14	4.68.97.33	ae-2-52.bbr2.NewYork1.Level3.net
15	64.159.1.130	ae-0-0.bbr2.SanJose1.Level3.net
16	4.68.123.73	ge-7-1.ipcolo1.SanJose1.Level3.net
17	63.215.192.174	unknown.Level3.net
18	216.171.128.239	border1.rap.sjc.transedge.com
19	216.171.128.82	f0-0.agg1.rap.sjc.transedge.com
20	**216.171.145.66**	**cust-router-redshift.transedge.com**
21	216.228.2.73	cs0-FE-0-0.corp.redshift.net
22	**216.171.145.66**	**cust-router-redshift.transedge.com**
23	216.228.2.73	cs0-FE-0-0.corp.redshift.net
24	**216.171.145.66**	**cust-router-redshift.transedge.com**
...	...	...

for 72.9.194.0/24. VSNL has a primary provider, Savvis (AS 3561), as the default route to access the Internet. So all traffic to 72.9.195.0/24 will form a persistent forwarding loop among these four ISPs. Table 9 shows the result of the traceroute, in which the forwarding loop involves 17 routers.

This example shows that although the misconfiguration occurs in a stub network, it may cause persistent forwarding loops that involve routers in multiple domains. The persistent forwarding loops can potentially be exploited by attackers to flood links in a backbone network in large ISPs or Tier-1 ASes.

5.3. Identifying persistent forwarding loops caused by neglecting to configure pull-up routes

Since neglecting to configure pull-up routes may cause persistent forwarding loops, it is important to understand how often such misconfigurations occur in the Internet. To gain some insight into this problem, we present a heuristic algorithm to identify persistent forwarding loops that are potentially caused by neglecting to configure pull-up routes.

Before we describe our algorithm, we revisit the example shown in Fig. 4. When the border router of Customer C is misconfigured with missing pull-up routes, the forwarding paths to subnets 18.1.1.0/24 and 18.1.3.0/24 in Customer C are different. That is, there is no persistent forwarding loop to the network 18.1.1.0/24, while there is a persistent forwarding loop to the network 18.1.3.0/24. Since Customer C only announces its prefix 18.1.0.0/16 instead of multiple subnets to its provider P, the differences in the forwarding paths to the two subnets in Customer C may not be visible from global routing tables. Therefore, it is hard to identify such misconfigurations in Customer C from global routing tables only. In general, there are some common properties with forwarding paths to a network when there is a misconfiguration resulting from neglecting to configure pull-up routes. Here we highlight the following three properties:

1. A network announces its contiguous IP addresses as one block to the Internet in which some

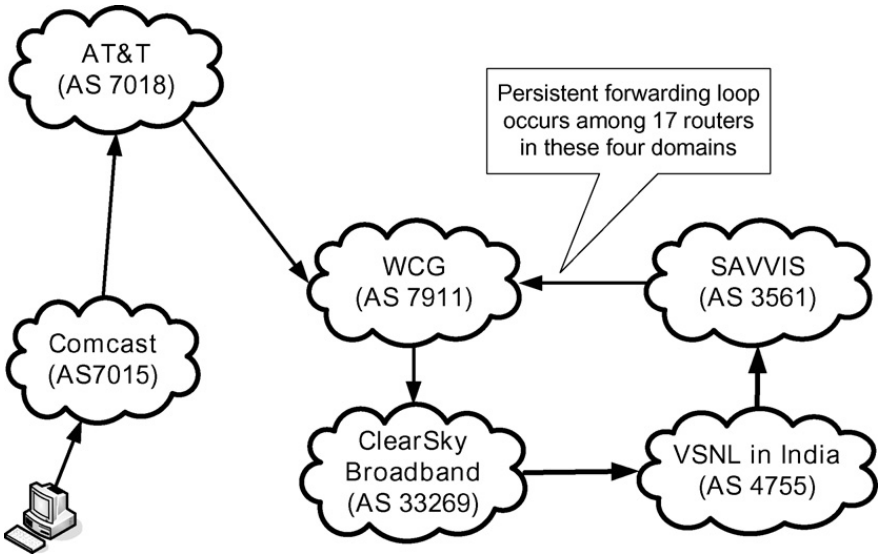


Fig. 7. Persistent forwarding loop across four domains.

Table 9
An example of traces that contain persistent forwarding loops among four domains

Hop	Router address	Router name	RTT (ms)
...	...	...	...
6	64.200.249.21	nycmny2wcx3-pos3-1.wcg.net	6.748
7	64.200.240.253	brvwil1wcx3-pos0-0-oc192.wcg.net	85.142
8	**64.200.240.121**	**dnvrco1wcx3-pos13-0-oc192.wcg.net**	**219.827**
9	64.200.106.37	dnvrco1wcx2-pos3-0-oc192.wcg.net	77.596
10	64.200.240.249	anhmca1wcx3-pos12-0-oc192.wcg.net	80.601
11	64.200.140.146	lsanca1wcx1-pos1-0-oc48.wcg.net	79.057
12	64.192.145.82	64-192-145-82.wcg.net	80.010
13	72.9.192.34	n/a	78.738
14	72.9.194.225	gw1.vsnl01.ind.clearskynet.net	355.391
15	203.197.173.99	colo-static.173.99.bng.vsnl.net.in	310.177
16	203.197.174.1	colo-static.174.1.bng.vsnl.net.in	350.636
17	202.54.2.69	mum-bgl-2nd-stm1.Bbone.vsnl.net.in	323.372
18	202.54.2.130	nny-mum-6th-stm1.Bbone.vsnl.net.in	314.158
19	208.192.178.117	gigabitethernet1-1.ig5.nyc4.alter.net	309.039
20	152.63.35.30	0.so-3-2-0.xl1.nyc4.alter.net	311.298
21	152.63.70.101	0.so-6-0-0.xl1.chi13.alter.net	316.545
22	152.63.69.177	pos6-0.gw1.chi13.alter.net	312.924
23	157.130.42.210	wcggigechi-gw.customer.alter.net	310.413
24	64.200.236.101	brvwil1wcx3-pos1-1-oc48.wcg.net	309.168
25	**64.200.240.121**	**dnvrco1wcx3-pos13-0-oc192.wcg.net**	**336.410**
26	64.200.106.37	dnvrco1wcx2-pos3-0-oc192.wcg.net	331.911
27	64.200.240.249	anhmca1wcx3-pos12-0-oc192.wcg.net	355.632
28	64.200.140.146	lsanca1wcx1-pos1-0-oc48.wcg.net	355.204
...	...	...	...

subsets of addresses experience persistent forwarding loops while some subsets do not.
2. From a given source address, forwarding paths to the subnets may share routers in the Internet, but diverge inside that network.
3. There is a misconfigured router on the forwarding paths. Some of these forwarding paths contain persistent forwarding loops, while some do not.

Based on the above discussions, we describe our algorithm for identifying persistent forwarding loops caused by missing pull-up routes in Fig. 8. For each persistent forwarding loop, first we find the longest prefix p in BGP routing tables in a default-free zone, where prefix p is a supernet of the shadowed prefix of the persistent forwarding loop. Then, we collect all traces to the addresses in prefix p, and denote them as $T(p)$. Third, we classify

For each persistent forwarding loop
1. find the prefix p in global BGP routing tables, where prefix p is a supernet of the shadowed prefix of the persistent forwarding loop
2. get all traces in D_A that are traced to the destination addresses in prefix p, which is denoted as $T(p)$
3. classify the traces $T(p)$ into two subsets, $T_1(p)$ and $T_2(p)$
   (a) subset one, $T_1(p)$
      — in which each trace contains the persistent forwarding loop
   (b) subset two, $T_2(p)$
      — in which each trace does not contain any forwarding loop
4. if there exist a trace t_1 in $T_1(p)$ and a trace t_2 in $T_2(p)$
   (a) if both t_1 and t_2 contain a router interface address r, and
   (b) if the address r appears in the persistent forwarding loop, and
   (c) if both forwarding paths from source address to router interface r in the traces t_1 and t_2 are the same

   then, the persistent forwarding loop is considered to be caused by neglecting to configure pull-up routes

Fig. 8. Identifying persistent forwarding loops caused by missing pull-up routes.

these traces into two subsets $T_1(p)$ and $T_2(p)$. Subset $T_1(p)$ contains the traces that have the persistent forwarding loop, and subset $T_2(p)$ contains traces that do not have any forwarding loops. Finally, we compare the traces in the two subsets. If a trace t_1 exists in $T_1(p)$ and a trace t_2 exists in $T_2(p)$, where both t_1 and t_2 contain a router interface r that appears in the persistent forwarding loop, and if both forwarding paths from source address to router interface r in t_1 and t_2 are the same, we attribute the persistent forwarding loop to misconfiguration neglecting pull-up routes.

In our study, we collect a prefix list P that contains all prefixes appearing in global BGP routing tables from RouteViews servers [12]. We then apply our algorithm in Fig. 8 on trace data D_A to estimate the percentage of persistent forwarding loops caused by neglecting to configure pull-up routes. Our algorithm yields an estimation that there are about 21% of persistent forwarding loops caused by configuration errors on pull-up routes.

Note that the condition 4(c) of our algorithm in Fig. 8 says that both forwarding paths from the source address to the shared router interface r in traces t_1 and t_2 should be exactly the same. However, due to the practice on ICMP rate limiting or filtering in routers, the traces could be slightly different in terms of observed router interfaces although they actually represent two identical forwarding paths in terms of traversed routers. For example, in Table 10, there are two traces t_1 and t_2 destined to 208.53.185.80 and 208.53.191.239, respectively. The trace to 208.53.185.80 contains a persistent forwarding loop, but the trace to 208.53.191.239 does not. Comparing these two forwarding paths, we believe that from hop 1 to hop 15, the forwarding path should be the same. However, in the real traces, the sequences of router interfaces are different because there are timeouts at hop 4 and hop 8 during traceroute. This might be caused by ICMP rate limiting or filtering, or even packet loss. In some other cases, two forwarding paths are almost the same except for one or two router interfaces. For example, the router interfaces at hop 16 in Table 10 are different in traces t_1 and t_2. This might be caused by load balancing in the Internet. These similar but non-identical forwarding paths are not treated as the "same path" in our algorithm.

Given the above limitations in our algorithm, we may fail to identify some persistent forwarding loops caused by misconfigurations on pull-up routes due to mismatches in router interfaces. Therefore,

Table 10
Similar forwarding paths in two traces

Hop	Trace t_1 to 208.53.185.80 (contains a persistent forwarding loop)	Trace t_2 to 208.53.191.239 (does not contain any forwarding loop)
1	10.221.248.1	10.221.248.1
2	68.87.159.17	68.87.159.17
3	68.87.37.66	68.87.37.66
4	**68.87.144.221**	*
5	68.87.144.217	68.87.144.217
6	68.87.144.77	68.87.144.77
7	68.87.144.197	68.87.144.197
8	*	**68.87.145.9**
9	12.118.88.9	12.118.88.9
10	12.122.81.14	12.122.81.14
11	12.122.10.106	12.122.10.106
12	12.123.6.69	12.123.6.69
13	154.54.11.205	154.54.11.205
14	154.54.2.237	154.54.2.237
15	66.28.6.142	66.28.6.142
16	**38.112.4.98**	**38.112.0.234**
17	**66.90.127.254**	**66.90.127.254**
18	**66.90.127.253**	208.53.191.239
19	**66.90.127.254**	
20	**66.90.127.253**	
21	**66.90.127.254**	
...	...	...

considering the practice of ICMP rate limiting and filtering, we relax the rules in our algorithm with the following two methods.

1. *Method A*: In this method, we relax the condition 4(c) of our algorithm in Fig. 8 by considering ICMP filtering, rate limiting and load balancing. The relaxation rules are shown in Fig. 9. The basic idea is to allow isolated mismatches in a series of router interfaces when comparing two traces. As shown in Fig. 9, for a mismatched router interface in traces t_1 and t_2, if the routers $r1$ and $r3$ that are *right before* and *right after* the mismatches are exactly the same in the two traces, we consider it a match. With this relaxation, about 39% persistent forwarding loops are identified as being caused by configuration errors on pull-up routes.
2. *Method B*: In this method, we apply further relaxations by ignoring the condition 4(c) of our algorithm in Fig. 8. That is, we check whether two traces t_1 and t_2 shared a router interface r, but do not require the two forwarding paths from the source address to r in traces t_1 and t_2 to be exactly the same. With this relaxation, we identified about 50% persistent forwarding loops as caused by configuration errors on pull-up routes.

Considering ICMP filtering or rate limiting:

Rule 1: two paths, (r1, *, r3) and (r1, *, r3), are considered the same.
Rule 2: two paths, (r1, *, r3) and (r1, r2, r3), are considered the same.

Considering load balancing:

Rule 3: two paths, (r1, ra, r3) and (r1, rb, r3), are considered the same.

Fig. 9. Relaxation rules in Method A.

Given the above relaxations in our algorithm, we may have over-estimated the percentage of loops caused by misconfigurations on pull-up routes compared with the original algorithm in Fig. 8, while our results still emphasize that it is quite common for network operators to neglect the configuration of pull-up routes.

There are some other limitations in our algorithm that may cause biases in our results. First, although misconfiguration on pull-up routes should exhibit the three properties in forwarding paths as we mentioned before, it does not imply that any forwarding behavior with such properties should be a result of missing pull-up routes. That is, the properties that our algorithms explored are the necessary conditions for misconfiguration on pull-up routes, but not sufficient conditions. Second, in our algorithm we compare the traces in two subsets in which the destination addresses are originated from the same prefix in BGP routing tables. However, when a network owns non-contiguous IP addresses and announces multiple prefixes to the Internet, the misconfigurations on pull-up routes with multiple prefixes cannot be detected by our algorithm. Finally, our algorithm does not resolve the alias of router interfaces. Even if a forwarding path does not contain any router interface appearing in the persistent forwarding loop, there is still a chance that one of them is indeed involved in the persistent forwarding loop due to router interface alias.

6. Conclusion and future work

In this paper we investigate vulnerability on flooding attacks by exploiting persistent forwarding loops. We emphasize that such vulnerability can be exploited from various locations, and can severely affect Internet connectivity to a significant number of network addresses. We further investigate the possible cause of persistent forwarding loops, and find that neglecting to configure pull-up routes is the common misconfiguration that causes them. We show that even if the misconfiguration occurs in a stub network, it may cause persistent forwarding loops involving routers in large ISPs or Tier-1 ASes and can potentially be exploited by attackers to flood links in a backbone network. These findings suggest that this vulnerability could be a critical threat to Internet security.

Although we have described the scope and possible cause of persistent forwarding loops, there are still several issues for future study along this line. Further measurement from different times over a longer period and different locations would be helpful to understand the properties of persistent forwarding loops. Also, we only identify 50% of persistent forwarding loops as caused by pull-up routes. We have not found any good explanations for the remaining 50%. We plan to investigate other possible causes that could lead to persistent forwarding loops in the future. Lastly we need to investigate ways to eliminate these persistent forwarding loops and minimize their impact on the Internet.

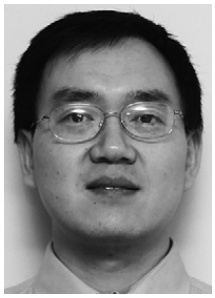
Acknowledgement

The authors thank Jennifer Rexford and anonymous reviewers for their insightful comments and constructive suggestions.

References

[1] American Registry for Internet Numbers. <http://www.arin.net/whois/arinwhois>.
[2] P. Francois, O. Bonaventure, Avoiding transient loops during IGP convergence in IP networks, in: Proceedings of IEEE INFOCOM, Miami, FL, vol. 1, March 2005, pp. 237–247.
[3] Z. Ge, D.R. Figueiredo, S. Jaiwal, L. Gao, Hierarchical structure of the logical Internet graph, in: Proceedings of SPIE ITCom 2001, Denver, CO, vol. 4526, August 2001, pp. 208–222.
[4] B. Halabi, Internet Routing Architectures, Cisco Press, 1997.
[5] U. Hengartner, S. Moon, R. Mortier, C. Diot, Detection and analysis of routing loops in packet traces, in: Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurement, Marseille, France, November 2002, pp. 107–112.

[6] Internet Protocol v4 Address Space. <http://www.iana.org/assignments/ipv4-address-space>.
[7] R. Mahajan, D. Wetherall, T. Anderson, Understanding BGP Misconfiguration, in: Proceedings of ACM SIGCOMM, Pittsburgh, PA, August 2002, pp. 3–16.
[8] Z.M. Mao, J. Rexford, J. Wang, R. Katz, Towards an accurate AS-level traceroute tool, in: Proceedings of ACM SIGCOMM, Karlsruhe, Germany, August 2003, pp. 365–378.
[9] V. Paxson, End-to-end routing behavior in the internet, IEEE/ACM Transactions on Networking (TON) 5 (5) (1997) 601–615.
[10] PlanetLab. <http://www.planet-lab.org/>.
[11] A. Sridharan, S. Moon, C. Diot, On the correlation between route dynamics and routing loops, in: Proceedings of the Internet Measurement Conference (IMC'03), Miami, FL, October 2003, pp. 285–294.
[12] University of Oregon Route Views Project. <http://www.routeviews.org/>.
[13] J. Xia, L. Gao, T. Fei, Flooding attacks by exploiting persistent forwarding loops, in: Proceedings of the Internet Measurement Conference (IMC'05), Berkeley, CA, October 2005, pp. 385–390.
[14] J. Xia, L. Gao, T. Fei, Flooding attacks by exploiting persistent forwarding loops, presented in the North American Network Operators' Group (NANOG36), Dallas, TX, February 2006.
[15] M. Zhang, C. Zhang, V. Pai, L. Peterson, R. Wang, PlanetSeer: Internet path failure monitoring and characterization in wide-area services, in: Proceedings of Sixth Symposium on Operating Systems Design and Implementation (OSDI'04), San Francisco, CA, December 2004, pp. 167–182.



Jianhong Xia received his B.S. degree and M.S. degree in Electrical Engineering from the University of Science and Technology of China in 1993 and 1996, respectively, his Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst in 2007. He worked with Huawei Technologies Co. Ltd (China) from 1996 to 1999. His research interests include Internet routing, network measurement and security.

Lixin Gao is an associate professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. She received her Ph.D. degree in computer science from the University of Massachusetts at Amherst in 1996. Her research interests include multimedia networking, and Internet routing and security. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS. She is an Alfred P. Sloan Fellow and received an NSF CAREER Award in 1999. She has served on number of technical program committees including SIGCOMM 2006, SIGCOMM2004, SIGMETRICS2003, and INFOCOM2004, and is on the Editorial Board of IEEE Transactions on Networking.

Teng Fei is a Ph.D. candidate at the University of Massachusetts, Amherst, and his research area is network measurement, overlay network, inter-domain routing and network security.