Advanced Routing Worm and Its Security Challenges

Cliff C. Zou, Don Towsley, Weibo Gong and Songlin Cai

SIMULATION 2006; 82; 75
DOI: 10.1177/0037549706065344

Advanced Routing Worm and Its Security Challenges

Cliff C. Zou
School of Electrical Engineering & Computer Science
University of Central Florida
Orlando, FL
czou@cs.ucf.edu

Don Towsley
Department of Computer Science
University of Massachusetts
Amherst, MA

Weibo Gong
Department of Electrical & Computer Engineering
University of Massachusetts
Amherst, MA

Songlin Cai
Parallogic Corporation
Sterling, VA

Most well-known worms, such as Code Red, Slammer, Blaster, and Sasser, infected vulnerable computers by scanning the entire IPv4 address space. In this article, the authors present an advanced worm called the "routing worm," which implements two new attacking techniques. First, a routing worm uses Border Gateway Protocol (BGP) routing tables to only scan the Internet-routable address space, which allows it to propagate three times faster than a traditional worm. Second, and more important, the geographic information of BGP routing prefixes enables a routing worm to conduct pinpoint "selective attacks" by imposing heavy damage to vulnerable computers in a specific country, company, Internet Service Provider, or autonomous system, without collateral damage done to others. Because of the inherent publicity of BGP routing tables, attackers can easily deploy routing worms, which distinguishes the routing worm from other "worst-case" worms. Compared to a traditional worm, a routing worm could possibly cause more severe congestion to the Internet backbone since all scans sent out by it are Internet routable (and can be dropped only at the destination local networks). In addition, it is harder to quickly detect a routing worm–infected computer since we cannot distinguish illegal scans from regular connections sent out from it without waiting for traffic responses. For high-fidelity Internet-scale worm simulations, through this routing worm study, the authors emphasize the importance of simulating failed worm scans and distinguishing nonroutable worm scans from routable scans. In order to defend against routing worms and all scanning worms, an effective way is to upgrade the current Internet from IPv4 to IPv6, although such an upgrade will require a tremendous effort and is still a controversial issue.

Keywords: Network security, routing worm, modeling

1. Introduction

Computer worms are malicious programs that self-propagate across a network, exploiting security or policy flaws in widely used services [1]. Most previous wide-spreading worms, such as Code Red, Slammer, Blaster, and Sasser [2], are scanning worms that find and infect vulnerable machines by probing IP addresses in the entire IPv4 Internet address space. How fast a worm can propagate is determined by many factors. Among them, three major factors could be improved by attackers:

- the number of initially infected hosts
- a worm's scan rate η, defined as the average number of scans an infected computer sends out per unit time

SIMULATION, Vol. 82, Issue 1, January 2006 75-85

DOI: 10.1177/0037549706065344

- a worm's hitting probability p, defined as the probability that a worm's scan hits any computer that is either vulnerable or already infected

"Hit-list worm," presented by Staniford, Paxson, and Weaver [3], exploits the first factor to improve a worm's propagation speed by containing a large number of IP addresses of vulnerable hosts in the worm code. The second factor, worm scan rate, is determined by the efficiency of a worm's code and also the network bandwidth. If attackers want to improve a worm's propagation speed, another effort is to increase the worm's hitting probability p (i.e., to waste fewer scans on obviously empty IP space).

To defend against Internet worm attacks, we need to anticipate and study how attackers will improve their attacking techniques. In this article, we present an advanced scanning worm called "routing worm," which increases its propagation speed by removing many empty IP addresses from its scanning space based on information of Border Gateway Protocol (BGP) *routable* addresses. We define two types of routing worms—one is based on "/8" prefix (x.0.0.0/8) address allocation; another is based on BGP routing prefixes. We call them "/8 routing worm" and "BGP routing worm," respectively. Without missing any potential target in the Internet, a /8 routing worm and a BGP routing worm can reduce their scanning space to 51.6% and 32.7% of the entire IPv4 address space, respectively. In this way, attackers can increase the spreading speed of their worms by a factor of two to three without adding much complexity to the worm codes.

The IP address information of BGP routing prefixes provides geographic information about which IP addresses belong to which country, company, Internet Service Provider (ISP), or autonomous system (AS). With such information, attackers could deploy a routing worm to selectively impose heavy damage to compromised hosts if they belong to a specific entity (country, company, ISP, or AS) and leave the compromised hosts belonging to others intact. Such a "selective attack" property makes a routing worm tremendously dangerous, considering the potential attacks initiated by terrorists, revengers, or business rivals.

Because of the inherent publicity of BGP routing tables, attackers can easily deploy a routing worm without much extra effort—this distinguishes the routing worm from other theoretical "worst-case" worms. In addition, compared to a traditional worm that scans the entire IPv4 space, a routing worm could possibly cause more congestion trouble to the Internet backbone and also makes it harder to quickly detect infected computers. We will explain these challenges in detail later in this article.

To defend against the threat of routing worms and all scanning worms, we show that upgrading the current IPv4 Internet to IPv6 is an effective way, although such an upgrade will require a tremendous effort and is still a controversial issue.

The rest of this article is organized as follows. Section 2 surveys related work. In section 3, we discuss how routing worms can use various types of IPv4 address information to improve their spreading speed. In section 4, we point out that attackers can use routing worms to conduct selective attacks based on the geographic information of IP addresses or BGP prefixes. Then, in section 5, we point out two additional challenges brought up by routing worms. In section 6, we emphasize two critical issues that must be considered in accurately simulating an Internet-scale worm propagation. In section 7, we present modeling and analysis of routing worms based on the uniform-scan worm model [4]. Then we propose upgrading IPv4 to IPv6 to defend against scanning worms in section 8. Section 9 concludes this article.

2. Related Work

At the same time that we proposed the "routing worm," Wu et al. [5] independently presented a "routable scan" strategy that is similar to the reducing scanning space idea of the routing worm. However, the routing worm presented in this article is not only a simple "routable scan" worm but also a worm that could be used by attackers to conduct selective attacks to a specific country or company (ISP, AS, etc.), which is more dangerous and important to attackers than simply improving a worm's propagation speed. Staniford, Paxson, and Weaver [3] presented several possible fast-spreading worms, such as "Warhol" worm and "hit-list" worm, right after the 2001 Code Red incident. Other researchers [3, 6-11] have provided major research work on how to model and analyze a worm's propagation under various situations.

Many people have studied how to derive the geographic information of ASs, ISPs, IP addresses, or domain names from public available information. The Skitter project provides detailed information of the AS number, name, longitude, and latitude for every AS in the Internet [12]. CAIDA [13] provides the mapping between AS number and the country it belongs to. Furthermore, there are location mapping commercial services, such as EdgeScape from Akamai [14] and the free IP-to-location service from Geobytes [15].

The Route Views project [16] and the Routing Information Service from RIPE NCC [17] provide detailed BGP routing information of the Internet. In 1997, Braun [18] first used BGP routing tables to determine the fraction of IP space that has been allocated. CAIDA also studied this issue in 1998 [19].

Some people have proposed upgrading IPv4 to IPv6 as a defense against scanning worms [7, 20, 21] but have not explained this issue in detail. Thus, most people have not paid attention to the inherent capability of IPv6 in preventing attacks from scanning worms.

3. Routing Worm: A Fast-Spreading Worm

The central idea of the spreading speed improvement of a routing worm is to make the worm's target finding more

efficient without ignoring any potential vulnerable computer in the Internet.

3.1 BGP Routing Worm

One simple way to reduce the scanning space is to use the information provided by BGP routing tables. Both the Route Views project [16] and RIPE NCC [17] provide complete snapshots of BGP routing tables several times per day. BGP routing tables contain all Internet-routable IP addresses. A *BGP routing worm* is an advanced worm that scans BGP-routable IP addresses to find potential targets to infect. In this way, the worm effectively reduces its scanning space without missing any target.

A BGP routing *prefix* is a chunk of IP addresses that have the same n most-significant bits in their addresses, where n is called *prefix length* for this prefix. For example, the prefix 10.0.0.0/8 has prefix length "8" and contains IP addresses ranging from 10.0.0.0 to 10.255.255.255, having the same first 8 bits equal to value 10. Because of multihoming, many prefixes in a BGP routing table overlap with each other—one prefix of shorter length contains all IP addresses in another prefix of longer length. For example, both 128.119.0.0/16 and 128.119.85.0/24 may appear in the BGP routing table, and the prefix 128.119.0.0/16 contains all IP addresses in the latter one.

To determine the percentage of IPv4 space that is BGP routable, we download BGP routing tables from Route Views [16], extract routing prefixes, and remove all overlapping prefixes that are contained by others. For the previous example of overlapping prefixes, we remove the second one, 128.119.85.0/24, from the BGP routing prefixes. In this way, we can calculate the percentage of allocated routable IPv4 space. We illustrate in Figure 1 how the utilization of IP space has evolved in the 8-year period from November 1997 to May 2005.

Although the number of computers connected to the Internet has increased greatly from 1997 to 2005, due to the usage of Classless Inter-Domain Routing (CIDR), Network Address Translation (NAT), and Dynamic Host Configuration Protocol (DHCP), the allocated routable IP space has not increased much. Figure 1 shows that 32.7% of IPv4 addresses were BGP routable in May 2005. By including the information of BGP routing prefixes, a BGP routing worm can reduce its scanning space by 67.3% without ignoring any potential vulnerable computer.

3.2 /8 Routing Worm

BGP routing tables in May 2005 contain more than 173,000 prefixes. After removing overlapping prefixes, a BGP routing worm still needs to contain about 78,000 prefixes. To avoid adding a big payload to a routing worm, attackers could possibly use IPv4 "/8" address allocation information instead of BGP routing prefixes.

The Internet Assigned Numbers Authority (IANA) provides public information about how the "/8" prefix (x.0.0.0/8) of IPv4 has been assigned [22]. Each "/8" prefix contains 2^{24} IP addresses, and there are 256 (2^8) "/8" prefixes in IPv4. By combining the IANA "/8" allocations with the information of BGP routing prefixes (BGP data from May 1, 2005), we find that 132 "/8" prefixes contain all BGP-routable IP addresses. In other words, from an attacker's point of view, a worm does not need to waste its scans on IP addresses belonging to the other 124 nonroutable "/8" prefixes.

A "/8 routing worm" is defined as an advanced worm that only scans those "/8" prefixes that contain BGP-routable addresses. According to the BGP data from May 2005, a /8 routing worm only needs to scan 51.6% of IPv4 space by adding a small 132-byte prefix payload.

In fact, Code Red II [23] has already used part of IANA address allocations to reduce its scanning space: if an IP address generated by a Code Red II worm belongs to 127.0.0.0/8 (loopback addresses) or 224.0.0.0/4 (16 "/8" multicast addresses [22]), then the worm skips that address and generates a new address to scan. In this way, Code Red II scans 93.4% of the entire IPv4 space (239 out of 256 "/8" address spaces).

3.3 Infection within Private Address Networks

Scanning BGP-routable space does not mean that a routing worm totally discards private IP space, such as 10.0.0.0/8 and 192.168.0.0/16 [22]. Because of limited IP address resources, many companies and organizations have used private IP addresses for their internal networks. In addition, most wireless local-area networks have used private IP addresses for their wireless clients. When compromising a vulnerable computer, a routing worm can first check whether the computer is using private IP addresses. If it is, a routing worm can scan both the private IP space and the BGP-routable space.

3.4 Storage Requirement for BGP Routing Information

After removing overlapping prefixes, a BGP routing worm still needs to contain about 78,000 prefixes, according to the BGP routing table from May 1, 2005. To spread out in the Internet quickly, a worm needs to have as small payload as possible to avoid the prolonged worm code transmission time. In addition, a worm with a smaller payload would spread faster by causing less severe network congestion. Therefore, the payload of a routing worm would be a bit large if it contains the remaining 78,000 BGP prefixes.

One way an attacker might try to reduce the routing prefix payload is to save the routing prefixes in a compact format, such as the following:

- For prefixes that are /8, 1 byte is used to store one prefix; prefixes that are between /9 and /16 use 2 bytes

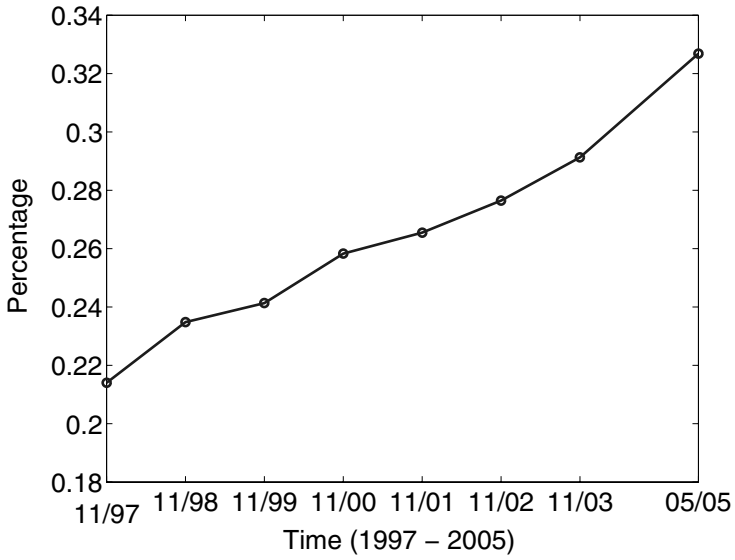


Figure 1. Percentage of Border Gateway Protocol (BGP) routable address space over the entire IPv4 space from 1997 to 2005 (data from Route Views project [16])

for each prefix; prefixes that are between /17 and /24 use 3 bytes for each prefix; and prefixes that are between /25 and /32 use 4 bytes for each prefix.

- Attackers can store BGP prefixes in the order of prefix lengths, as shown in Figure 2.

By using the above storage format, the 78,000 routing prefixes carried by a BGP routing worm can be stored in a 220-KB payload. Because the storage method shown in Figure 2 is already tight, using compression program such as gzip or Winzip only reduces the payload less than 30 KB. Therefore, attackers cannot gain much benefit by using a compressed payload.

3.5 Routing Worm Based on Prefix Aggregation

A BGP routing worm scans a potentially smaller space than a /8 routing worm, but its routing prefix payload, as shown above, is much larger. To have a good trade-off between the size of the scanning IP space and the payload requirement for a routing worm, attackers can aggregate BGP routing prefixes. Here, *aggregation* means that many BGP prefixes are combined into one that has a shorter prefix length by adding the empty IP space between those original ones.

For example, the two prefixes 128.119.254.0/24 and 128.119.255.0/24 can be aggregated into one prefix, 128.119.254.0/23, without adding any IP addresses; they can also be aggregated into the prefix 128.119.0.0/16 by adding IP addresses from 128.119.0.0 to 128.119.253.255.

In this way, the newly generated prefix covers all the IP space in those original prefixes. Through aggregation, a routing worm would need to scan a larger IP space but store fewer prefixes in its payload.

One simple aggregation method is to aggregate all prefixes that have prefix lengths longer than n to be "$/n$" prefixes ($8 \leq n \leq 32$), which is called "$/n$ aggregation." If $n = 32$, no prefixes need to be aggregated, and a BGP routing worm is derived; if $n = 8$, a /8 routing worm is derived.

Figure 3 shows the aggregation impact on a routing worm's scanning space and prefix payload. For clarity, we only show the aggregation results from "/16" aggregation to "/8" aggregation in this figure. It shows that, as a routing worm aggregates more BGP prefixes together, it increases its scanning space while reducing the size of its payload.

By using prefix aggregation, attackers have the freedom to choose a suitable "$/n$" aggregation according to their needs or the desired spreading properties of a routing worm.

4. Routing Worm: A Selective Attack Worm

By considering IP address information, a routing worm not only increases its propagation speed but also can exploit such information to conduct dangerous selective attacks, which is a more important property to attackers. "Selective attack" means that hackers or terrorists can selectively impose heavy damage to vulnerable computers in a specific country, company, ISP, or AS with little collateral damage done to others.

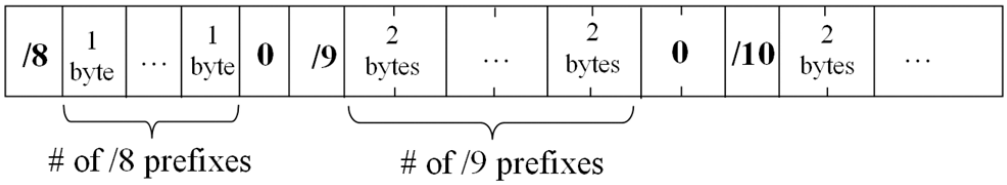


Figure 2. One possible storage format of Border Gateway Protocol (BGP) routing prefixes: "/8" and "/9" represent the prefixes /8 and /9, and they occupy 1 byte each; "0" is used to indicate the ending of prefixes—it occupies the same number of bytes as the prefixes before it

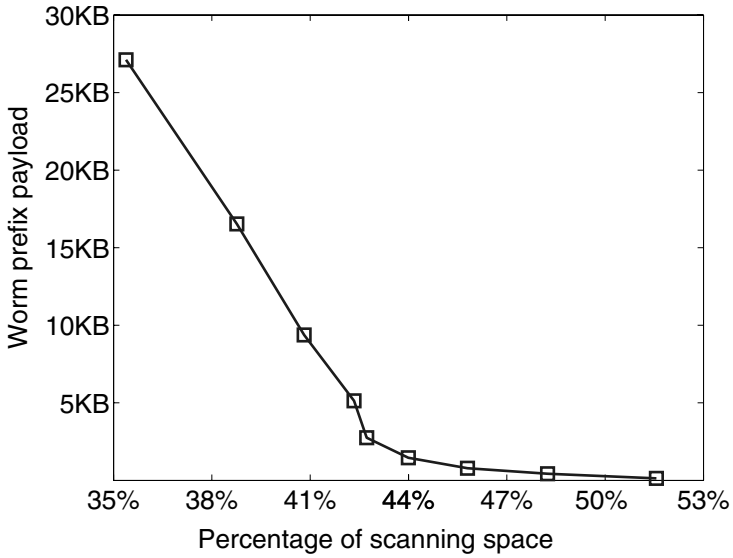


Figure 3. Prefix aggregation impact on a routing worm's scanning space and prefix payload (each point from left to right represents "$/n$" aggregation, where $n = 16, 15, \cdots, 8$, respectively)

4.1 Selective Attack Based on IP Geographic Information

IANA provides limited information about who owns a "/8" network [22]. For example, 214.0.0.0/8 and 215.0.0.0/8 are allocated to the U.S. Department of Defense, 56.0.0.0/8 is allocated to the U.S. Postal Service, 43.0.0.0/8 is allocated to Japan Inet, and so on [22]. Such information can be possibly used by attackers in their /8 routing worm if they want to attack these specific targets.

In addition, from IANA public data, attackers are able to know what "/8" addresses are allocated to a region. For example, 23 "/8" prefixes are allocated to the American Registry for Internet Numbers (ARIN), which is the Regional Internet Registry (RIR) of North America, South America, the Caribbean, and sub-Saharan Africa [22].

Meanwhile, BGP routing tables provide detailed information about what AS owns a specific network prefix. Since many people have studied how to derive geographic information from BGP routing prefixes [12, 13], hackers, revengers, or terrorists can use routing worms to conduct pinpoint heavy attacks to vulnerable computers in a specific country, company, ISP, or AS with little collateral damage to others.

Attackers can program a routing worm to exhibit different behaviors based on the location of the compromised computers. For example, if a compromised computer belongs to a specific country or company, the routing worm can impose heavy damage to this computer; otherwise, the compromised computer will be simply used as a stepping stone to scan and infect others without being destroyed. For another example, attackers can program a routing worm to have a higher scanning preference for IP prefixes belonging to a specific target—this "target preference" scanning

method is an extension of the "local preference" used by Code Red II [2].

4.2 Selective Attack: A Simple but General Attacking Idea

In fact, "selective attack" is a simple but very general attacking idea for any large-scale spreading virus or worm. Viruses or worms can use any information they retrieve from compromised computers to conduct selective attacks. Such information of a compromised computer includes the computer's IP address, time zone, operating system, installed software, CPU, memory, network connection type and speed, and so on. For example, a worm can selectively impose heavy damage on compromised computers if they have installed illegal Windows operating systems, a specific peer-to-peer file-sharing program, or video cards from a specific manufacturer.

Besides inflicting damage, attackers can also use the "selective attack" idea to improve a worm's spreading speed. For example, on any compromised computer, Code Red always generates 100 threads to scan and infect others simultaneously [24]. However, some compromised computers that have a small-size memory or a slow network connection cannot support those 100 threads without crash; on the other hand, many compromised computers that have powerful CPU, large memory, and high connection speed may be able to support thousands of threads generated by the worm. Therefore, attackers can program a worm to generate a different number of scanning threads based on computer resources to speed up the worm's overall spreading speed.

By using "selective attack," attackers have more freedom to define viruses or worm behaviors; they also can obtain more control over their viruses or worms. In fact, a primitive selective attack has already been implemented by Code Red II—the worm generates 300 threads if a compromised computer runs non-Chinese Windows and 600 threads if the computer runs Chinese Windows [23].

5. Other Security Challenges from a Routing Worm

Besides its fast-spreading speed and selective attack properties, as explained in the above two sections, a routing worm imposes two additional challenges to the Internet and our defense systems. In this section, we discuss these two challenges in detail.

5.1 Network Congestion Challenge to Internet Backbone

When an ordinary scanning worm is transformed into a routing worm, it may cause more severe congestion to the backbone of the Internet.

Since about 2/3 of IPv4 space is not BGP routable, around 2/3 of scan packets sent out by an ordinary worm target IP space that is not Internet routable. These packets will be quickly dropped at "default-free" routers[1] before entering the backbone links of the Internet. Thus, around 2/3 of worm scan traffic will not appear on the Internet backbone.

On the other hand, all scans sent by a routing worm are BGP routable and, hence, will travel across the Internet backbone and reach the routers of the destination ASs or local networks. Therefore, a routing worm (especially a bandwidth-limited routing worm) will cause more severe congestion trouble to the Internet backbone than current scanning worms that scan the entire IPv4 space.

For example, Slammer worm has caused severe congestion in many parts of the Internet [26]. If this worm writer had changed the worm code to be a routing worm, it would possibly have caused severe congestion to the entire Internet infrastructure instead of congestion in many local-area networks.

5.2 Worm Detection Challenge

A routing worm makes it harder to *quickly* detect and then quarantine internal infected computers in an enterprise network.

As explained in Staniford [9], to defend an enterprise network against a fast worm attack, the defense system of the enterprise network must be able to identify and then quarantine an internal infected computer as quickly as possible before the worm spreads out. One general detection method is to detect the illegal traffic sent out by an infected computer due to the random scans generated by a scanning worm [9, 27, 28]. For an ordinary worm that scans the entire IPv4 space, because a large percentage of the worm's random scans target nonroutable address space, we can quickly detect an internal infected computer based on its outgoing illegal connection destinations without waiting for the traffic response.

On the other hand, all scans sent out by a routing worm–infected computer are Internet routable. Thus, we have to wait a while for the traffic response of those scans (such as TCP timeout or ICMP error messages from routers) to determine that these connection requests are abnormal. For the defense of a fast-spreading worm such as the Slammer worm, such a detection time difference might be critical for shutting down the worm infection process before it is too late.

6. Simulation Considerations

High-fidelity Internet-scale worm simulation is a very important way to understand how a worm spreads in the Inter-

1. A default-free router is a router that "actively decides where to send packets with a destination outside the AS to which the router belongs, and not forward it, by default, to another router" [25].

net, how to conduct effective quarantine and defense, what is required to contain a worm's impact within a certain level, and so on. Through this routing worm study, we find out two important issues that must be carefully considered to conduct an accurate Internet-scale worm simulation.

6.1 Simulation of Failed Worm Scans

Suppose a "successful scan" is defined as a worm's connection attempt that finds a vulnerable computer as the target; a "failed scan" is defined as a worm's connection attempt that targets an empty IP address or a not vulnerable computer (to this worm). An important fact of an Internet-scale scanning worm is that the worm will generate many more failed scans than successful scans.

For example, Code Red infected 360,000 computers, and it scanned the entire Internet to find targets [24]. In this way, each Code Red scan has the probability $p = 360,000/2^{32} = 0.0000838$ to hit a vulnerable target, which means that, on average, a compromised computer needs to send out

$$1/p = 11,930$$

scans to hit *one* vulnerable target (the target may have already been infected by others). Even if the Code Red is transferred to a BGP routing worm, a compromised computer still needs to send out $1/p \times 0.327 = 3900$ scans to hit one target.

Therefore, in an Internet-scale worm simulation, we cannot just simulate the network impact and infection delay time of those successful scans. We must also accurately simulate the huge number of failed scans since they could be the major cause for link congestions and router failures.

6.2 Different Treatment of Nonroutable Worm Scans and Routable Scans

As pointed out in section 5.1, a traditional worm that scans the entire Internet spends 2/3 of its scans targeting nonroutable IP space. These scans will be discarded at the first default-free routers, and hence they will not appear in the Internet backbone. The other 1/3 of scans will pass through the Internet backbone and reach the destination local networks—they either reach the target computers or are discarded by the routers at the destination local networks because they target empty IP addresses.

Therefore, when simulating failed worm scans, we need to distinguish scans targeting nonroutable space from scans targeting BGP-routable space. Scans targeting BGP-routable space will contribute to the possible congestion at three places: the Internet backbones, the source local networks, and the destination local networks. On the other hand, scans targeting nonroutable space only contribute to the possible congestion at the source local networks.

7. Routing Worm Propagation Modeling and Analysis

In our previous article [29], we presented a uniform-scan worm model that is described by worm propagation parameters:

$$\frac{dI_t}{dt} = \frac{\eta}{\Omega} I_t (N - I_t), \tag{1}$$

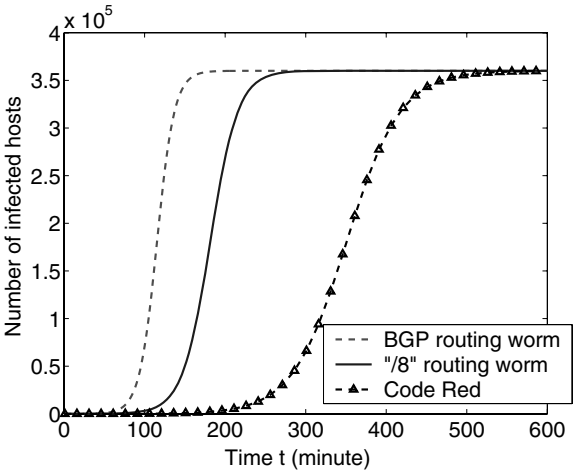
where I_t is the number of infected hosts at time t, and N is the total number of vulnerable hosts in the system before the worm spreads out. At $t = 0$, I_0 hosts are infected, and the remaining $N - I_0$ hosts are vulnerable. η is the worm's average scan rate, and Ω is the size of the worm's scanning space.

If a routing worm uniformly scans its scanning space and has the same average scan rate as a traditional worm, then according to (1), a routing worm will propagate faster due to its smaller scanning space Ω. To show how much faster a routing worm can propagate, we use Code Red as the example of a traditional worm, which has a scan rate $\eta = 358$ per minute and a vulnerable population $N = 360,000$ [7]. We assume that there are $I_0 = 10$ initially infected hosts. Figure 4(a) shows the numbers of infected hosts I_t of the Code Red worm, a /8 routing worm, and a BGP routing worm as functions of time t, respectively. It shows that by using IP routing information, routing worms clearly increase their spreading speed.

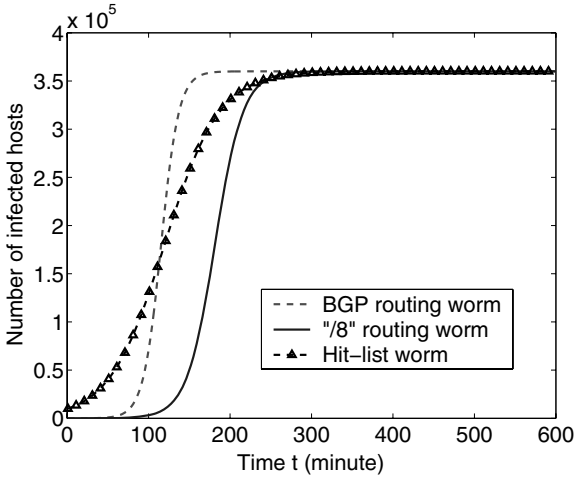
Staniford, Paxson, and Weaver [3] introduce a "hit-list" worm that has an address list of a large number of vulnerable hosts in the Internet. Since a hit-list worm can infect all vulnerable hosts in its hit-list within a few seconds [3], we ignore this hit-list infection time and assume that a hit-list worm begins to propagate with a large number of initially infected hosts, where I_0 equals the size of the hit list. When a hit-list worm uniformly scans the Internet after its hit-list scanning phase, its propagation can be modeled by (1) with $\Omega = 2^{32}$.

To study the propagation differences between a hit-list worm and routing worms, Figure 4(b) compares a BGP routing worm, a /8 routing worm, with a hit-list worm that has a hit list of 10, 000 vulnerable hosts and the same scan rate $\eta = 358$/min as Code Red. This figure shows that the hit-list worm can infect a larger number of hosts in a short time, but its infection growth rate is smaller than routing worms.

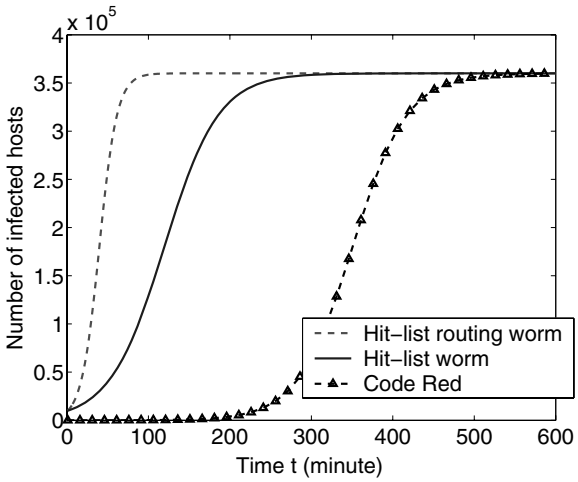
A hit-list worm and a routing worm try to improve their spreading speed through two different approaches. These two approaches do not conflict with each other and can be easily combined together to generate a new worm, called a "hit-list routing" worm, that has both a large number of initially infected hosts and a fast propagation speed. Figure 4(c) shows the propagation of a hit-list routing worm, which has a 10,000 hit list and the BGP routing prefixes. Compared with a traditional worm and an ordinary hit-list worm, the hit-list routing worm spreads out much faster.



(a) Comparison between two types of routing worms and Code Red



(b) Comparison between two types of routing worms and a hit-list worm



(c) Comparison between a hit-list routing worm, a hit-list worm, and Code Red

Figure 4. Worm propagation comparisons. (a) Comparison between two types of routing worms and Code Red. (b) Comparison between two types of routing worms and a hit-list worm. (c) Comparison between a hit-list routing worm, a hit-list worm, and Code Red.

The famous *Warhol* worm presented in Staniford, Paxson, and Weaver [3] is a hit-list worm that uses a "permutation scan" instead of a uniform scan. The permutation scan provides a form of coordination among infected hosts to avoid multiple scanning on the same IP addresses [3], which cannot be modeled by the uniform-scan worm model (1). Due to the coordination mechanism, the Warhol worm propagates faster than a uniform-scan hit-list worm after most vulnerable hosts have been infected (as shown in Figure 3 in Weaver [20]). However, a routing worm and the original Code Red can also deploy the same permutation scan instead of a uniform scan without any problem. If a routing worm and Code Red implement the same permutation scan as a Warhol worm, these three worms will have a similar propagation relationship to that shown in Figure 4 (although the pattern of propagation curves will change slightly, as shown in Figure 3 in Weaver [20]).

Staniford, Paxson, and Weaver [3] also present a *flash* worm that contains IP addresses of all vulnerable hosts in the worm's hit list. A flash worm can infect all vulnerable computers in the Internet within tens of seconds [3]. However, it is very hard or impossible to collect up-to-date IP addresses of all vulnerable hosts in the global Internet, especially for computers that do not advertise their addresses

(such as SQL database servers attacked by Slammer or the ISS security products attacked by Witty worm [30]). Therefore, flash worms exist in theory and are not likely to be generated by attackers in an Internet-scale attack (although it is possible for attackers to use a flash worm to attack a local-area network).

Due to its tiny payload requirement, a "/8 routing worm" might be used by attackers in their future bandwidth-limited worms. A "bandwidth-limited worm" is a worm that fully uses the link bandwidth of an infected host to send out infection traffic. For example, SQL Slammer is a bandwidth-limited worm with an average scan rate $\eta = 4000$ scans/second [26]. Because Slammer is a UDP-based worm that puts the complete worm code into one single UDP packet, the BGP routing worm idea is not realistic for this worm. Each UDP infection packet sent out by Slammer is 404 bytes [26]. If the worm author transformed Slammer into a /8 routing worm, which is called a "routing Slammer worm," the UDP infection packet would be 520 bytes (by adding a 132-byte prefix payload). After transforming into a /8 routing worm, the routing Slammer worm would have an average scan rate $\eta = 4000 \times 404/536 = 3015$ scans/second. Figure 5 shows the worm propagation of the original Slammer and the new routing Slammer worm as functions of time (the other parameters are $N = 100,000$, $I(0) = 10$, the same as what is used in Zou et al. [7]).

8. Defense against Routing Worms: Upgrading IPv4 to IPv6

It is very hard to prevent attackers from generating a routing worm due to the following two reasons: (1) both IANA "/8" allocations and BGP routing tables are publicly available information that is difficult or impossible to hide from attackers, and (2) a routing worm is very easy for attackers to implement—much easier than the hit-list worm presented in Staniford, Paxson, and Weaver [3]. Once attackers obtain BGP routing prefixes, they can use the same BGP data for all scanning worms to attack various vulnerabilities. On the other hand, to program a hit-list worm, attackers need to collect a hit list of vulnerable computers and have to repeat such work for different vulnerabilities. Such a hit list is especially hard to collect for vulnerable hosts that do not advertise their addresses (e.g., Windows SQL servers attacked by Slammer). In addition, many computers change their IP addresses frequently. Because of the real threat coming from a routing worm, and also because of the serious challenges brought up by a routing worm as introduced in section 5, we must find a way to prevent a routing worm from quickly spreading out.

A routing worm increases its propagation speed by reducing its scanning space. Figure 4 shows how much faster a routing worm can propagate when the worm reduces its scanning space by only half to two-thirds. Therefore, if we use the same principle to dramatically increase a worm's scanning space, we can significantly slow its propagation speed. For this reason, we believe that an effective defense against routing worms and all scanning worms is to upgrade the current IPv4 Internet to IPv6—the vast address space of IPv6 (its BGP tables include no prefixes longer than /64) can prevent a worm from spreading through scanning.

IPv6 has dramatically increased IP space from 32-bit addresses to 128-bit addresses. Because of this huge IP address space, IPv6 implements a hierarchical addressing theme where the smallest network has 2^{64} IP addresses (with prefix /64) [31, 32]. In other words, the smallest network in IPv6 BGP routing tables contains the number of IP addresses equal to that of *4 billion* IPv4 Internet.

Some people might think that allocating such a big address space for a smallest network wastes too much of the IP resource. Actually, it does not. Suppose there are *1000 billion* people on earth; then, on average, each person can own 2.3 million, the smallest networks (/64) mentioned above for unicast usage.

Attackers can still use BGP routing tables to program a routing worm. However, they are not able to know address allocation information inside any /64 network from BGP routing tables since the longest prefix in IPv6 BGP routing tables is /64. A local network might use a smaller address space for internal address allocation, but such information will not show up in BGP routing tables and thus is not known to attackers. It does not matter if the local address allocation within a /64 space follows some specific rules—as long as attackers do not know the rules, attackers cannot shrink their worm's scanning space without port-scanning beforehand.[2]

Even one single /64 network in IPv6 will have sufficient IP space to defeat scanning worms. Suppose there are $N = 1,000,000$ vulnerable hosts in one single /64 network and a worm has a scan rate $\eta = 100,000$/second with $I_0 = 1000$ initially infected hosts. If the worm only scans and infects hosts in this /64 network, then $\Omega = 2^{64}$. Based on (1), the worm will need to spend *40 years* to infect half of the vulnerable hosts in this single /64 network.

Of course, upgrading IPv4 to IPv6 is not the omnipotent solution for defending all kinds of worm attacks. It is only useful for defending worms that find victims by random scanning, such as Code Red, Slammer, Blaster, Sasser, and Witty worm. In addition, IPv6 is still a controversial issue, and there are many important economic and technical details to be solved before we can upgrade the current IPv4 to IPv6.

9. Conclusions

In this article, we present a new advanced scanning worm called "routing worm." Based on BGP routing prefix

2. For this reason, computers in a local network should not use their default IEEE-802 MAC addresses for the lowest 48-bit in their IPv6 addresses.

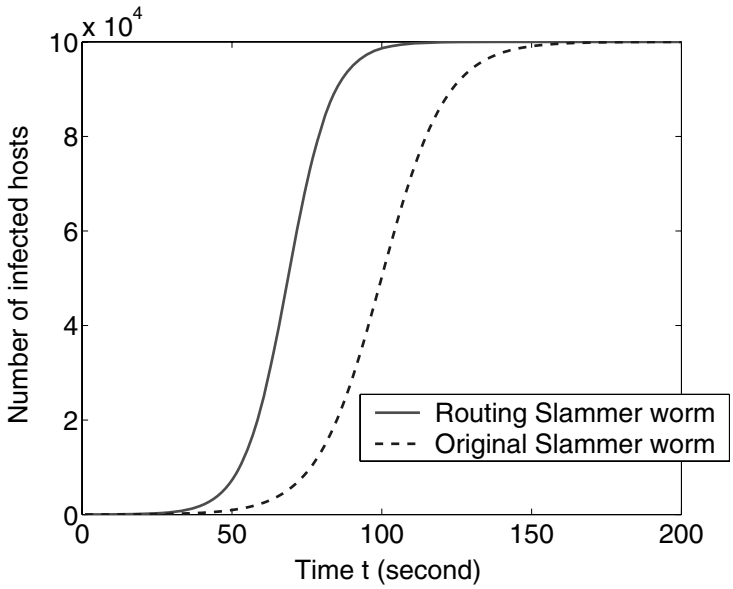


Figure 5. Worm propagation comparison of the original Slammer with the /8 routing worm transformed from Slammer

information, a routing worm not only propagates faster but also is able to conduct pinpoint selective attacks to a specific country, company, ISP, or AS. Because of the inherent publicity of BGP routing tables, attackers can easily deploy routing worms in the future. Compared to a traditional worm, a routing worm could possibly cause more severe congestion trouble to the Internet backbone, making it harder to quickly detect infected computers. An effective way to defend against routing worms and all scanning worms is to upgrade the current IPv4 to IPv6, although such an upgrade will require a tremendous effort and is still argued by many people.

10. Acknowledgments

This work was supported in part by ARO contract DAAD19-01-1-0610, NSF grant EEC-0313747, EIA-0080119, ANI-0085848, and CNS-0325868.

11. References

[1] Weaver, N., V. Paxson, S. Staniford, and R. Cunningham. 2003. A taxonomy of computer worms. In *Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM'03)*, pp. 11-8.

[2] CERT. CERT/CC advisories. www.cert.org/advisories/

[3] Staniford, S., V. Paxson, and N. Weaver. 2002. How to own the Internet in your spare time. In *Proceedings of USENIX Security Symposium*, pp. 149-67.

[4] Zou, C. C., D. Towsley, and W. Gong. 2003. On the performance of Internet worm scanning strategies. Umass ECE Department.

[5] Wu, J., S. Vangala, L. Gao, and K. Kwiat. 2004. An efficient architecture and algorithm for detecting worms with various scan techniques. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS'04)*.

[6] Zou, C. C., W. Gong, and D. Towsley. 2002. Code Red worm propagation modeling and analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02)*, pp. 138-47.

[7] Zou, C. C., L. Gao, W. Gong, and D. Towsley. 2003. Monitoring and early warning for Internet worms. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 190-9.

[8] Chen, Z., L. Gao, and K. Kwiat. 2003. Modeling the spread of active worms. In *Proceedings of the IEEE INFOCOM*, pp. 1890-1900.

[9] Staniford S. Forthcoming. Containment of scanning worms in enterprise networks. *Journal of Computer Security*.

[10] Nicol, D., and M. Liljenstam. 2004. Models of Internet worm defense. In *IMA Workshop 4: Measurement, Modeling and Analysis of the Internet*. www.ima.umn.edu/talks/workshops/1-12-16.2004/nicol/talk.pdf

[11] Kesidis, G., I. Hamadeh, and S. Jiwasurat. 2005. Coupled Kermack-McKendrick models for randomly scanning and bandwidth- saturating Internet worms. In *Proceedings of 3rd International Workshop on QoS in Multiservice IP Networks (QoS-IP)*, pp. 101-9.

[12] CAIDA. 2003. *Visualizing Internet topology at a macroscopic scale*. www.caida.org/analysis/topology/as_core_network

[13] CAIDA. 2003. *IPv4 BGP geopolitical analysis*. www.caida.org/analysis/geopolitical/bgp2country

[14] Akamai Service: EdgeScape. www.akamai.com/en/html/services/edgescape.html

[15] Net World Map project: IP address locator tool. www.geobytes.com/IpLocator.htm?GetLocation

[16] University of Oregon Route Views Project. www.routeviews.org

[17] RIPE NCC Routing Information Service. www.ripe.net/ris

[18] Braun, H. W. 1997. *BGP-system usage of 32 bit Internet address space*. http://moat.nlanr.net/IPaddrocc

[19] CAIDA. 1998. *IPv4 address space utilization*. www.caida.org/outreach/resources/learn/ipv4space

[20] Weaver, N. 2001. *Warhol worms: The potential for very fast Internet plagues*. www.cs.berkeley.edu/~nweaver/warhol.html

[21] Warfield, M. H. 2003. Security implications of IPv6. White paper, Internet Security Systems, Inc.
[22] IANA. 2004. *Reserved IPv4 addresses*. www.cidr-report.org/v6/reserved-ipv4.html
[23] eEye Digital Security. 2001. *CodeRedII worm analysis*. www.eeye.com/html/Research/Advisories/AL20010804.html
[24] Moore, D., C. Shannon, and J. Brown. 2002. Code-Red: A case study on the spread and victims of an Internet worm. In *Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurement*, pp. 273-84.
[25] Antony, A., and H. Uijterwaal. 1999. *Routing Information Service R.I.S. design note*. www.ripe.net/projects/ris/Notes/ripe-200/
[26] Moore, D., V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. 2003. Inside the Slammer worm. *IEEE Magazine on Security and Privacy* 1 (4): 33-9.
[27] Jung, J., S. E. Schechter, and A. W. Berger. 2004. Fast detection of scanning worm infections. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pp. 59-81.
[28] Weaver, N., S. Staniford, and V. Paxson. 2004. Very fast containment of scanning worms. In *Proceedings of 13th USENIX Security Symposium*, pp. 29-44.
[29] Zou, C. C., D. Towsley, and W. Gong. Forthcoming. On the performance of Internet worm scanning strategies. *Journal of Performance Evaluation*.
[30] Shannon, C., and D. Moore. 2004. *The spread of the Witty worm*. www.caida.org/analysis/security/witty/
[31] Hinden, R., and S. Deering. 2003. *RFC-3513: Internet Protocol Version 6 (IPv6) addressing architecture*. http://www.rfc-archive.org/getrfc.php?rfc=3513
[32] Hinden, R., S. Deering, and E. Nordmark. 2003. *RFC-3587: IPv6 global unicast address format*. http://www.rfc-archive.org/getrfc.php?rfc=3587

Cliff C. Zou *is an Assistant Professor in the School of Electrical Engineering & Computer Science at the University of Central Florida, Orlando.*

Don Towsley *is a Professor in the Department of Computer Science at the University of Massachusetts, Amherst.*

Weibo Gong *is a Professor in the Department of Electrical & Computer Engineering at the University of Massachusetts, Amherst.*

Songlin Cai *is a Senior Software Engineer in Parallogic Corporation, Sterling, VA.*