Routing Worm: A Fast, Selective Attack Worm based on IP Address Information

Cliff C. Zou*, Don Towsley†, Weibo Gong*, Songlin Cai*
*Department of Electrical & Computer Engineering
†Department of Computer Science
Univ. Massachusetts, Amherst
Technical Report: TR-CSE-03-06

Abstract

Most well-known Internet worms, such as Code Red, Slammer, and Blaster, infected vulnerable computers by scanning the entire Internet IPv4 space. In this paper, we present a new scan-based worm called "routing worm", which can use information provided by BGP routing tables to reduce its scanning space without ignoring any potential vulnerable computer. In this way, a routing worm can propagate twice to more than three times faster than a traditional worm. In addition, the geographic information of allocated IP addresses, especially BGP routing prefixes, enables a routing worm to conduct fine-grained selective attacks: hackers or terrorists can selectively impose heavy damage to vulnerable computers in a specific country, an Internet Service Provider, or an Autonomous System, without much collateral damage done to others. Routing worms can be easily implemented by attackers and they could cause considerable damage to our Internet. Since routing worms are scan-based worms, we believe that an effective way to defend against them and all other scan-based worms is to upgrade IPv4 to IPv6 — the vast address space of IPv6 (2^{64} IP addresses for a single subnetwork) can prevent a worm from spreading through scanning.

I. Introduction

Computer worms are programs that self-propagate across a network exploiting security or policy flaws in widely-used services [14]. The easy access and wide usage of the Internet make it a primary target for the propagation of worms. Since the first well-known Morris worm [10] in 1988, attackers have continuously developed worms. Today, our computing infrastructure is more vulnerable [9] than ever before. In 2001, Code Red, Code Red II, and Nimda showed us how vulnerable our networks are [7][22][32]. Code Red infected more than 360,000 IIS servers within one day, causing millions-of-dollar loss to our society [42] — it began a new wave of global-scale propagation of worms. On January 25, 2003, SQL *Slammer* was released and quickly spread throughout the Internet [6]. Because of its super fast scan rate, Slammer infected more than 90% of the vulnerable computers on the Internet within 10 minutes [6]. In addition, the large amount of scan packets sent out by Slammer caused a global-scale denial of service attack to the Internet; many networks across Asia, Europe, and America were effectively shut down for several hours [43]. Only a half-year later, the Blaster worm appeared and infected more than 200,000 computers within a couple of hours on August 11, 2003 [31].

Code Red, Code Red II, Nimda, Slammer, and Blaster have created a new wave of global-scale fast spreading worms. All of them are scan-based worms that find and infect target machines by testing any IP address in the entire Internet IP address space. Nimda deployed several spreading mechanisms [22]. However, its fastest spreading mechanism was still based on scanning. As scan-based worms, their scanning mechanisms differ slightly from each other: Code Red, Nimda, and Slammer uniformly generate IP addresses to scan; Blaster scans IP addresses sequentially from a randomly generated IP address, or

from the first address of its "Class C" subnetwork (x.x.x.0/24) [31]; Code Red II uses "local preference" scanning — it has a higher probability to generate an IP address within the same Class B or the same Class A subnetwork than a random IP address [32].

It seems that attackers have chosen "scan-based" infection as the primary mechanism for the propagation of worms. The scan-based spreading mechanism exhibits the following attractive properties:

- Attackers do not need to collect any information about where vulnerable computers are.
- It's easier to program worms that propagate through random scanning than to program worms that need to find targets on compromised computers by themselves.
- The current IPv4 Internet address space is relatively small; little time is required for a worm to find a vulnerable target by random scanning. Thus a scan-based worm can have very fast propagation speed.

How fast a worm can propagate is determined by many factors: the vulnerable population size, the connection speed between computers, etc. Among these factors, there are three major factors that can be improved by attackers:

(1). The number of initially infected hosts.
(2). A worm's scan rate η, defined as the number of scans an infected host sends out in a unit time.
(3). A worm's hitting probability p, the probability that a worm's scan hits a target computer that is vulnerable at the beginning of the worm's propagation.

Suppose there are N vulnerable computers on the Internet before the spread of a worm, then the worm's hitting probability is $p = N/2^{32}$ if it scans the entire IPv4 space. If attackers want to improve worms' propagation speed, one effort is to reduce the scanning space of worms.

In order to defend against Internet worm attacks, we need to anticipate and study how attackers will improve their attacking techniques. In this paper, we propose a scan-based worm called the "routing worm", which increases its propagation speed by reducing its scanning space without missing any potential target. We define two types of routing worms — one based on Class A (x.0.0.0/8) address allocations, and the other based on BGP routing tables. We call them "BGP routing worm" and "Class A routing worm", respectively. Without missing any potential target, the Class A routing worms can reduce their scanning space to 45.3% of the entire IPv4 address space; the BGP routing routing worms can reduce their scanning space to only about 28.6% of the entire IPv4 address space. In this way, attackers can increase the spreading speed of routing worms by a factor of two or three without adding much complexity to their worms.

The IP address information, especially the BGP routing table, provides geographic information about which IP addresses belong to which country, company, Internet Service Provider (ISP), or Autonomous System (AS). With such geographic information, hackers or terrorists can selectively attack a specific target in various ways: a routing worm can impose heavy damage to all compromised hosts if they belong to a specific entity (country, company, ISP, or AS) and leave the hosts belonging to others intact; a routing worm can also exhibit a higher scanning preference for a specific entity in order to infect as many vulnerable machines as possible within that entity.

Since routing worms are scan-based worms and can be easily implemented by ordinary attackers, we believe that an effective way to defend against such worms and all scan-based worms is to upgrade IPv4 to IPv6 — the vast address space of IPv6 (2^{64} IP addresses for a single subnetwork) can prevent a worm from spreading through scanning.

The rest of this paper is organized as follows. Section II surveys related work. In Section III, we discuss how routing worms can use various types of IPv4 address information to improve their spreading speed. In Section IV, we point out that attackers can use routing worms to conduct selective attack based on geographic information of IP addresses or BGP prefixes. Section V briefly introduces the simple epidemic model, which is the worm propagation model we use in this paper. In Section VI, we present modeling and analysis of routing worms based on simple epidemic model. Then we propose to upgrade IPv4 to IPv6 to defend against scan-based worms in Section VII. In the end, Section VIII concludes this paper.

II. Related work

The closest work to ours is [11], which presented the "hit-list" worm right after the 2001 Code Red incident. A hit-list worm has a "hit-list" that contains IP addresses of a large number of potential vulnerable computers. The worm first scans and infects computers on the hit-list, and then continues to spread through random scanning. During the hit-list scanning phase, the worm can infect all vulnerable computers in the hit-list in a very short period of time since the worm does not need to waste scans on others. In a sense, a hit-list worm begins spreading with a large number of initially infected hosts. The routing worm tries to increase the hitting probability p by restricting the scanning IP address space.

Many people have studied how to derive the geographic information of ASes, ISPs, IP addresses, or domain names from public available information. The Skitter project at CAIDA provides detailed information of the AS number, name, longitude and latitude for every AS in the Internet [34]. In [33], CAIDA provides the mapping between AS number and the country it belongs to. NetGeo is a tool that maps IP addresses, domain names, and AS numbers to their geographic locations [8]; it mainly infers location from *Whois* records. [25] lists seven approaches to obtain the geographic location of an IP address. Padmanabhan and Subramanian present three distinct techniques to determine the geographic location of an IP address [12]. They point out that the most promising technique among those three techniques is the *GeoCluster*, which combines partial host-to-location mapping information and BGP prefix information to infer the location of a target host. Finally, there are location mapping commercial services, such as EdgeScape from Akamai [41] and the free IP-to-location service from Geobytes [35].

The Route Views project [30] and the Routing Information Service from RIPE NCC [37] provide detailed BGP routing information of the Internet. In 1997, Braun first used BGP routing tables [17] to determine the fraction of IP space that has been allocated. CAIDA also studied this issue in 1998 [28].

In the area of virus and worm modelling, Kephart, White and Chess of IBM performed a series of studies from 1991 to 1993 on viral infection based on epidemiology models [3][4][5]. Staniford *et al.* used the classical simple epidemic model to model the spread of Code Red right after the July 19th Code Red incident [11]. This model matches well the increasing part of the observed data. Zou *et al.* present a "two-factor" worm model that considers the effect of human countermeasures and the congestion caused by worm scan traffic [15]. Chen *et al.* present the discrete-time version of the worm model that considers the patching and cleaning effect during a worm's propagation [1]. Some of these worm propagation models are simple epidemic model, the others are extended from the simple epidemic model.

Some people have proposed upgrading IPv4 to IPv6 as a defense against scan-based worms [26][13][16], but have never explained this issue in detail. Thus most people have not paid attention to the capability of IPv6 in preventing scan-based worms.

III. Routing Worm: A Fast Spreading Worm

In this section, we will explain why routing worms spread more rapidly than traditional worms. The central idea of a routing worms is to reduce the IP space that it scans without ignoring any potential vulnerable computer.

A. BGP Routing Worm based on BGP Routing Table

One simple way to reduce the scanning space is to use the information provided by Border Gateway Protocol (BGP) routing tables. BGP is the inter-autonomous system routing protocol. An Autonomous System (AS) is a network or a group of networks under a common administration and with a common routing policy [29]. When BGP is used between ASes, the protocol is referred to as External BGP (EBGP). In this paper, we consider the routing tables of EBGP routers. Both the Route Views project [30] and the Routing Information Service from RIPE NCC [37] provide complete snapshots of BGP routing tables collected from EBGP routers all over the world several times per day. The BGP routing table contains all Internet routable IP addresses that have actually been allocated by the Internet Assigned Number Authority

(IANA) or various Internet Registries. Therefore, routing worms can only scan IP addresses contained in BGP routing tables to effectively reduce their scanning address space without missing any target.

A BGP routing table contains allocated routable IP address prefixes. A *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, which have the same first 8 bits equal to value 10. Because of multi-homing, many prefixes in BGP routing table overlap with each other — one prefix contains all of the 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 prefix.

To determine the percentage of IPv4 space that has been allocated, we downloaded the BGP routing tables from both Route Views [30] and RIPE NCC [37]. We extracted routing prefixes from both BGP routing tables, combined them together, and removed all overlapping IP prefixes that have longer prefix lengths. 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 Fig. 1 how the utilization of IP space has evolved in the last six years (1997 to 2003).

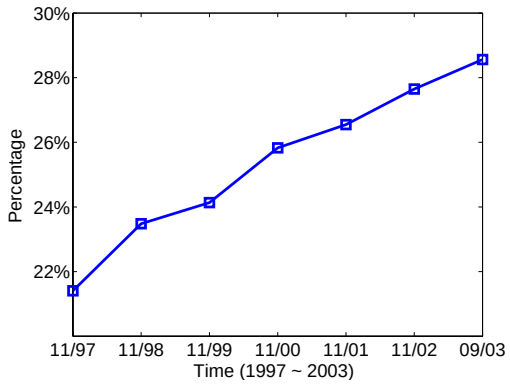


Fig. 1. The percentage of allocated routable IP space from 1997 to 2003 (over the whole IPv4 address space)

In November 1997, about 21.4% of the IP space had been allocated. After six years on September 22, 2003, only about 28.6% of the IPv4 space has been allocated (not including special used IP space that are non-routable, such as the 10.0.0.0/8 private network addresses prefix). Although the number of computers connected to the Internet has increased greatly in the last six years, 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.

Fig. 1 shows that currently about 28.6% of IPv4 addresses are allocated and routable. By including the information of BGP routing prefixes, routing worms can reduce their scanning space by 71.4% without ignoring any potential vulnerable computer.

B. Class A Routing Worm based on Class A Address Allocations

Currently, the number of BGP routing prefixes exceeds 140,000. Thus routing worms based on BGP prefixes will have a large payload. If attackers want to improve the propagation speed of traditional worms without adding too much payload to their worms, there is an alternative: using IPv4 Class A address allocation data.

The Internet Assigned Number Authority (IANA) takes charge of allocating IP addresses for the Internet. It provides public information about how Class A (x.0.0.0/8) addresses of IPv4 have been allocated or assigned to Regional Internet Registries (RIR) for further fine-grained allocation [39]. Each Class A space

contains 2^{24} IP addresses (these addresses have the same 8 most-significant bits). Thus there are 256 (2^8) Class A allocations in IPv4.

IANA provided the latest update of Class A allocation on April 5, 2003 [39]. However, this update suffers some problems, some of which are stated in [40]. For this reason, we use the BGP routing prefixes from September 22, 2003 to verify the Class A allocation update of IANA. By combining the IANA Class A allocations with the information of BGP routing prefixes, we derive the more accurate information about Class A address allocations and list them in Table I.

TABLE I

IPV4 CLASS A ADDRESS ALLOCATIONS BY IANA

Usage	Number of Class A prefixes
Multicast	16
IANA reserved	107 [1]
Non-routable	2 [2]
Allocated but inactive	15 [3]
Companies or organizations	28
ARIN	16
RIPE NCC	10
APNIC	11
Other Internet Registries	51

As shown in Table I, for the BGP routing table on September 22, 2003, there are 14 Class A addresses assigned to companies but not active according to the BGP routing table (we list them in the Appendix). All Internet accessible computers will have IP addresses that belong to the second group shown in Table I. In other words, from an attacker's point of view, worms need not waste their scan efforts on IP addresses that belong to the first group shown in Table I. If a worm originally propagates by scanning the entire IPv4 address space (256 Class A space), now with such Class A information, it only needs to scan 116 Class A address space, which contributes 45.3% of the entire IPv4 space.

To further reduce their scanning space, routing worms based on Class A address allocation can consider several additional reserved IP prefixes [39]: 128.0.0.0/16, 169.254.0.0/16, 172.16.0.0/12, 191.255.0.0/16, etc. However, because these IP address blocks contain much fewer addresses than a single Class A address block, the overall scanning space will not be reduced much by removing these small IP address blocks. In fact, Code Red II has already used part of IANA address allocations to reduce its scanning space [32]: if the IP address generated by a Code Red II worm belongs to 127.0.0.0/8 (Class A loopback addresses) or 224.0.0.0/4 (16 Class A multicast addresses), or is the same as the local machine's IP address, then the worm skips that address and generates a new IP address to scan. In this way, Code Red II scans only 93.4% of the entire IPv4 space (239 out of 256 Class A address space).

C. Storage requirement for BGP routing information

Table II lists the detailed information of the BGP routing prefixes on September 22, 2003 (extracted from both routing tables downloaded from Route Views [30] and RIPE NCC [37]). A routing worm would contain the remaining prefixes after overlapping prefixes are removed. The table shows that removing overlapping prefixes can dramatically reduce the storage requirement of routing worms — they only need to store 62053 prefixes instead of the original 140602 prefixes.

[1]105 Class A are shown as "IANA Reserved"; the other 2 that originally belong to "Joint Technical Command" are back to IANA on March 1998 [40].

[2]The two "non-routable" Class A are 10.0.0.0/8 and 14.0.0.0/8. The 10.0.0.0/8 is used for private network IP addresses. 14.0.0.0/8 is shown as "Public-Data Network" in [39], but it is inactive and we cannot find it in the BGP routing table.

[3]According to BGP routing table on September 22, 2003, 14 Class A addresses that are allocated to companies by IANA are inactive (see Appendix); another one, 191.0.0.0/8, is assigned to "various registries" but not used.

TABLE II

BGP ROUTING PREFIXES (SEPTEMBER 22, 2003)

Prefix	Number of prefixes (original BGP table)	Number of prefixes in routing worms
/8	18	18
/9	4	2
/10	6	6
/11	12	12
/12	55	50
/13	97	85
/14	265	255
/15	470	446
/16	7450	7070
/17	1833	1230
/18	3223	2308
/19	8625	6394
/20	9098	5770
/21	6766	2443
/22	9929	3505
/23	11602	3799
/24	74335	28612
/25∼/31	6814	48
Overall	140602	62053

In order to spread out on the Internet quickly, routing worms need to have as small payload as possible to avoid possible network congestion, which can be caused by a large number of infected computers transferring the big payload of routing worms. The remaining 62053 BGP prefixes still contribute a large payload for a routing worm.

one way an attacker can reduce the routing prefix payload of a routing worm is by saving the routing prefixes in the following format:

- For prefixes that are /8, one byte is used to store one prefix; prefixes that are /9 to /16 use two bytes for each prefix; prefixes that are /17 to /24 use three bytes for each prefix; prefixes that are /25 to /32 use four bytes for each prefix.
- Attackers can store BGP prefixes in the order of prefix lengths as shown in Fig. 2.

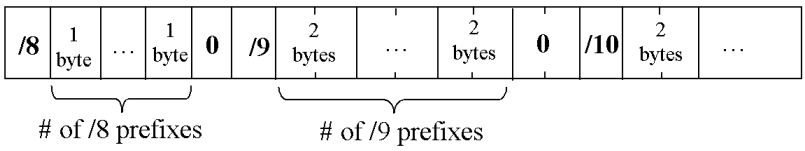


Fig. 2. One possible storage format of BGP routing prefixes: "/8" and "/9" represent the prefixes /8 and /9 and they occupy one byte each; "0" is used to indicate the ending of prefixes — it occupies the same number of bytes as the prefixes before it.

Hence the 62053 routing prefixes shown in Table II can be stored in a 175KB payload. Because the storage method shown in Fig. 2 is already tight, using compression program such as gzip or Winzip to compress the payload only reduces its size from 175KB to 154KB. Therefore, attackers cannot gain much benefit by using a compressed payload. In order to do so, they would have to program their own decompression algorithm into their routing worms.

D. Routing worm based on aggregated BGP prefixes

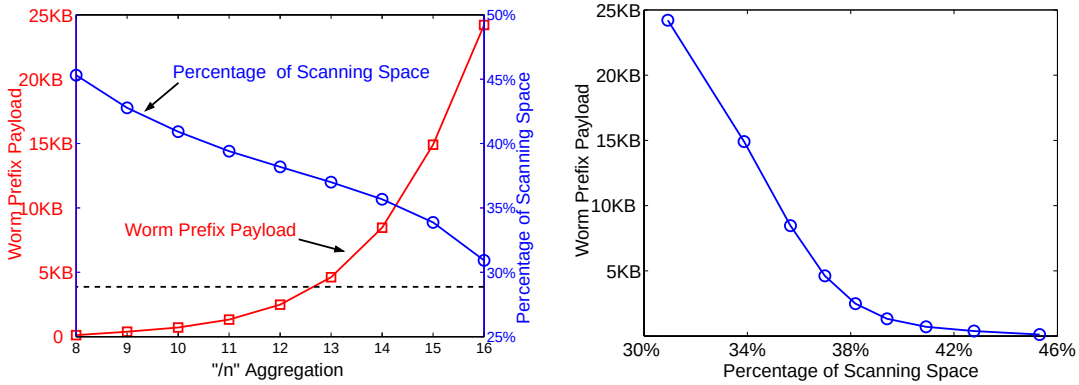
Until now, we have discussed two types of routing worms: BGP routing worm and Class A routing worm. The BGP routing worm scans a potentially smaller space than the Class A routing worm, but its

routing prefix payload is much larger. In fact, we can treat these two routing worms as two extreme cases for taking advantage of public information on the allocated IP addresses.

In order to have a finer trade off between the size of the scanning IP space and the payload requirement of routing worms, attackers can aggregate BGP routing prefixes. Here "aggregation" means that attackers can combine many BGP prefixes into one prefix that has a shorter prefix length by adding the empty space between those original prefixes. 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. Through aggregation, routing worms need to store fewer number of prefixes in their 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 can be called "$/n$ aggregation". If $n = 32$, no prefixes need to be aggregated; if $n = 8$, the BGP prefixes will be aggregated to be the same as the Class A allocation shown in Table I (overall 116 routable Class A).

Fig. 3 shows the aggregation impact on routing worms' 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 routing worms aggregate more BGP prefixes together, they increase their scanning space while reducing further their payload. For example, if a routing worm uses "/16" aggregation on BGP routing prefixes, the worm increases the scanning space from the original 28.6% to 30.9% of the IPv4 space, while reducing the prefix payload dramatically from the original 175KB to 24KB.



a. "/n" aggregation ($n = 8 \sim 16$) b. Payload and scanning space trade-off

Fig. 3. Prefix aggregation impact on a routing worm's scanning space and prefix payload (In the left-hand figure, the left Y-axis represents routing worms' prefix payload; the right Y-axis represents the percentage of scanning space over the whole IPv4 address space. The dash horizonal line shows the percentage of scanning space when routing worms directly use BGP prefixes without aggregation. In the right-hand figure, each point from left to right represents "/n" aggregation $n = 16, 15, \cdots, 8$, respectively.)

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

IV. Routing Worm: A Selective Attack Worm based on IP Geographic Information

By considering IP address information, routing worms not only increase their propagation speed, but can also use such information to conduct selective attacks. "Selective attack" means that attackers or terrorists can selectively impose heavy damage to vulnerable computers in a specific country, an Internet Service Provider, or an Autonomous System with little collateral damage done to others.

A. Selective attack based on Class A address allocations

As shown in Table I, the Class A address allocation provides limited information about where the IP addresses of a Class A network belong to. Attackers can easily know what companies or organizations

own those 28 Class A addresses. For example, attackers can know that 214.0.0.0/8 and 215.0.0.0/8 are allocated to "US-DOD", which is the US Department of Defense; 56.0.0.0/8 is allocated to "US Postal Service"; 43.0.0.0/8 is allocated to "Japan Inet", etc [39].

In addition to those 28 Class A addresses, attackers can know what other Class A addresses are allocated to a region. For example, there are 16 Class A addresses belong to American Registry for Internet Numbers (ARIN), which is the Regional Internet Registry (RIR) of North America and Sub-Sahara Africa [39].

B. Selective attack based on BGP routing prefixes

IANA Class A address allocations only provide very limited information about IP addresses. On the other hand, BGP routing tables provide detailed information about what Autonomous System (AS) owns a specific network prefix. Many people have studied how to derive the geographic information of ASes, ISPs, IP addresses, or domain names from public available data [25][12][35][34][33][8]. CAIDA provides a detailed mapping between AS number and which country an AS belongs to [33]. With such geographic information and BGP routing prefixes, attackers or terrorists can use routing worms to conduct pinpoint attacks on vulnerable computers in a specific country, Internet Service Provider, or Autonomous System with little collateral damage to others.

Attackers can program routing worms to exhibit different behavior based on the location of the compromised computers. For example, if a compromised computer belongs to a specific country or ISP, a routing worm can impose heavy damage to this computer; if the computer belongs to other countries or ISPs, the routing worm will impose no damage and just use the compromised computer as a stepping stone to scan and infect other vulnerable computers.

Attackers can also use geographic information of IP addresses for other kinds of selective attacks. For example, if attackers want to infect as many hosts as possible in one country, AS, or ISP, they can program routing worms to have higher scanning preference for IP prefixes that belong to the specific target. This "target preference" scanning method can be thought as an extension of the "local preference" scanning method used by Code Red II and Blaster. On the other hand, attackers can also program routing worms not to infect any vulnerable computers in one specific country, AS, or ISP by removing the corresponding BGP prefixes from the routing worms.

To selectively attack a target, attackers may need to program routing worms to scan and infect vulnerable computers at locations other than the target. One reason is that the number of vulnerable computers in the target may be limited and worms may require a large number of infected computers to scan and spread out quickly. Another reason is that if a vulnerable computer in the target is immediately destroyed right after it is compromised — to prevent it from being cleaned by security staffs later — this computer probably cannot continue to scan and infect others. In this case, routing worms need other infected computers to spread out from continuously.

In fact, a primitive selective attack has already been implemented by Code Red II — the worm will generate 300 threads if a compromised computer runs non-Chinese Windows and 600 threads if the computer runs Chinese Windows [32].

C. Selective attack based on aggregated BGP prefixes

When attackers use prefix aggregation, many geographically separated prefixes will be aggregated together. Thus if attackers directly aggregate BGP prefixes, they may not be able to conduct selective attacks anymore.

This problem can easily be handled by attackers. Attackers can first separate the original BGP prefixes into two groups: one group contains all prefixes that belong to the specific target, another group contains all other prefixes. Then attackers can aggregate prefixes in each group separately. In this way, although the aggregation efficiency decreases a little — the prefix payload will be larger than the original aggregation that has all prefixes as one group — attackers can still conduct selective attacks based on aggregated prefixes.

D. Selective attack: a simple but general attacking idea

Routing worms can conduct selective attacks based on geographic information of IP addresses of compromised computers. In fact, from an attacker's point of view, "selective attack" is a simple but very general idea useful for any large-scale spreading virus or worm. The "selective attack" idea in routing worms can be easily extended to rely on other information besides IP addresses — viruses or worms can use any information they can get from compromised computers to conduct selective attacks. Such information of a compromised computer includes its IP address, Operating System, installed software, CPU power, the volume of memory, network connection type and speed, computer hardware, etc.

For example, attackers probably can tell whether a computer is installed with a legal or illegal Operating System (many Microsoft patches cannot be installed on illegal Windows XP machines). Then attackers can selectively impose heavy damage on compromised computers that belong to either one of these two groups. For another example, attackers can selectively impose heavy damage to compromised computers that have a specific software installed, such as an anti-virus software, or a peer-to-peer file swapping software. For an example of hardware attack, attackers can impose heavy damage to compromised computers that are manufactured by a specific company, or compromised computers that have a CPU, or a motherboard manufactured by a specific company.

If attackers do not want to conduct malicious actions on compromised computers and just want their viruses or worms to spread out as quickly as possible, they can also benefit from using the "selective attack" idea. For example, on a compromised computer, Code Red generates 100 threads to scan and infect others simultaneously [7]. However, different infected computers on the Internet have very different CPU power, volume of memory, and network connection speed. For many computers that have either small size of memory or slow network connection speed, those 100 threads generated by Code Red may consume too many resources, perhaps causing those computers to crash. On the other hand, many compromised computers that have high-power, large memory, and high connection speed can support more threads than the 100 threads generated by the worm. To optimize the propagation speed of Code Red like worms, attackers can let their worms to generate different number of scanning threads based on computer resources.

Therefore, by using "selective attack", attackers can have more freedom to define viruses or worms behaviors; they also can obtain more control over their viruses or worms.

V. Simple Epidemic Model Introduction

Routing worms are general scan-based worms that impose no constraint for what scanning mechanism should be used. Hence they can use uniform scan, local preference scan, sequential scan, or any other technique. For simplicity, in this paper we study worms that use uniform scan, the method used by Code Red and Slammer. For such a random scanning method, we can use the simple epidemic model to study a worm's propagation. The results can be extended to other random scanning methods with minor modification. For example, for other random scanning methods used by worms, we can change the simple epidemic model to other suitable worm models, such as the Kermack-Mckendrick epidemic model [2], or the two-factor worm model [15].

We first briefly introduce the simple epidemic model. The model assumes that each host resides in one of two states: susceptible or infectious. The classical simple epidemic model for a finite population is [2]:

$$\frac{dI_t}{dt} = \beta I_t[N - I_t], \tag{1}$$

where I_t is the number of infected hosts at time t; N is the size of vulnerable population; and β is referred to as the pairwise rate of infection in classical epidemic studies [2]. At $t = 0$, I_0 hosts are infectious while the remaining $N - I_0$ hosts are all susceptible.

The epidemic model (1) has the analytical solution [2]:

TABLE III

NOTATIONS IN THIS PAPER

Notation	Definition
N	Number of vulnerable hosts under consideration
I_t	Number of infected hosts at time t
β	Pairwise rate of infection in epidemic model
α	Infection rate per infected host, $\alpha = \beta N$
α_0	Infection rate of worms that scan the whole IPv4 space
α_1	Infection rate of routing worms based on BGP routing table
α_2	Infection rate of routing worms based on Class A address allocation
η	Average scan rate per infected host
Ω	The size of IP address space that a worm scans
p	Probability that a worm's scan hits anyone among the population N

$$I_t = \frac{I_0 N}{I_0 + (N - I_0)e^{-\beta N t}} \tag{2}$$

Define

$$\alpha = \beta N \tag{3}$$

as the *infection rate*, the average number of vulnerable hosts that can be infected per unit time by one infected host during the early stage of a worm's propagation [16]. Zou *et al.* provide the following formula [16]:

$$\alpha = \eta \cdot \frac{N}{\Omega} \tag{4}$$

where η is the average worm scan rate; Ω is the size of a worm's scanning IP address space. If we assume that a worm scans the whole IPv4 space to propagate, then $\Omega = 2^{32}$ as shown in [16].

Define α_0 to be the infection rate of traditional worms that scan the entire IPv4 address space; α_1 to be the infection rate of a BGP routing worm; α_2 to be the infection rate of a Class A routing worm. If we assume that routing worms do not change their scan rate η when augmenting their payload with the information on BGP prefixes, then according to (4), such routing worms will have faster infection rate α_1 or α_2 than traditional worms have:

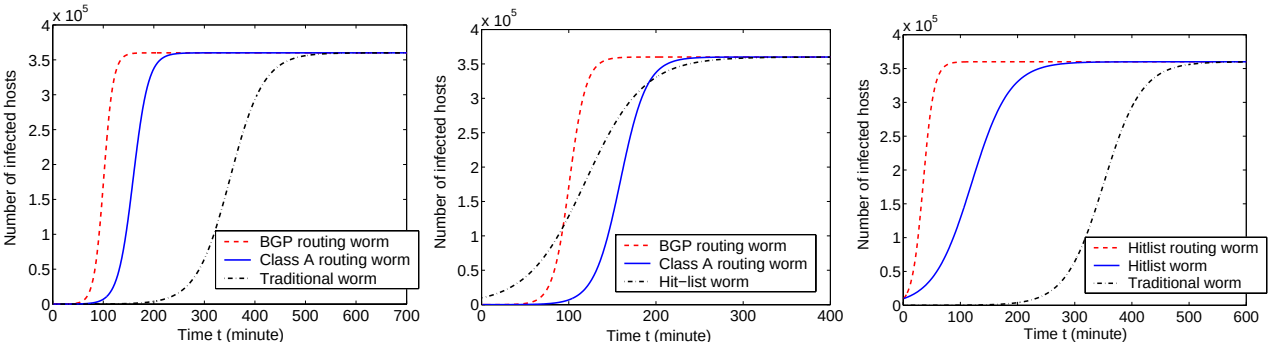
$$\alpha_1 = 1/0.286 \cdot \alpha_0 = 3.5\alpha_0 \tag{5}$$

$$\alpha_2 = 256/116 \cdot \alpha_0 = 2.21\alpha_0 \tag{6}$$

VI. ROUTING WORM PROPAGATION MODELING AND ANALYSIS

A. Routing worm: a fast spreading worm

If routing worms have the same scan rate as traditional worms, then according to (6) and (5), routing worms will propagate faster than traditional worms. To show how much faster routing worms can propagate, we use the simple epidemic model (1) to study worm propagation. Here we use Code Red as the example of a traditional worm, which scans the entire IPv4 space, has a scan rate $\eta = 358$ per minute, and a vulnerable population $N = 360,000$ [16]. We also assume there are $I_0 = 10$ initially infected hosts. In this case, Code Red has infection rate $\alpha_0 = \eta N/2^{32} = 0.03$ per minute according to (4). A BGP routing worm has scan rate $\alpha_1 = 0.105$ according to (5); a Class A routing worm has scan rate $\alpha_2 = 0.0663$ according to (6). Fig. 4(a) shows the number of infected hosts I_t of these three worms as functions of time t. It shows that by using IP space information, routing worms greatly increase their spreading speed.



a. Comparison between two types of routing worms and the Code Red worm

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

c. Comparison between a hit-list routing worm, an ordinary hit-list worm, and the Code Red worm

Fig. 4. Worm propagation comparisons (note that the time scale in (b) is smaller than the time scale in (a))

In this study, a routing worm begins with the same number of initially infected hosts as the traditional worm, but has a larger infection rate α. On the other hand, a "hit-list" worm [11] has the same infection rate as the traditional worm but begins with a large number of initially infected hosts — we assume that all vulnerable hosts in the hit-list are infected at the beginning of a hit-list worm's propagation [11]. Fig. 4(b) compares these two routing worms with a hit-list worm with a hit-list of $10,000$ vulnerable hosts, i.e., the hit-list worm has $I_0 = 10,000$ and has the original infection rate $\alpha_0 = 0.03$.

Fig. 4(b) shows that the hit-list worm can infect a larger number of hosts in a short time, but it has slower spreading speed than a routing worm. A hit-list worm and a routing worm try to improve the spreading speed of worms through two different approaches — a hit-list worm tries to increase the number of initially infected hosts I_0, while a routing worm tries to increase the worm's infection rate α by reducing its scanning space. Therefore, these two approaches do not exclude each other and can be easily combined together to generate a new worm, a "hit-list routing" worm, that has both a large number of initially infected hosts and fast propagation speed. Fig. 4(c) shows the propagation of a hit-list routing worm, which has a 10,000 hit-list and the BGP routing information. Compared with the worm propagation of a traditional worm and an ordinary hit-list worm, the hit-list routing worm spreads out faster.

If an attacker transforms the Slammer worm into a routing worm by adding the additional 116-byte Class A allocations to the original 376-byte Slammer Code, the scan rate of the routing worm will not decrease much. In the Slammer worm case, adding a hit-list in the worm may not increase the worm's spreading speed — the large hit-list payload may dramatically decrease the worm's scan rate η. If we assume that the new Slammer routing worm has the same scan rate as Slammer (4000 scans per second), Fig. 5 shows the propagation comparison between the Class A routing worm and the Slammer worm [4] ($N = 100,000$, $I_0 = 10$, $\eta = 4000$ per second as in [16]).

B. "Divide and conquer" propagation study

For large-scale worm propagation, attackers will try to design a worm with as small a payload as possible — a large payload will reduce the spreading rate of a worm, and may cause congestion to the network and thus slow down the worm's propagation.

Besides the "/n aggregation" method mentioned in Section III, there is an alternative to reduce the payload of BGP routing worms: BGP routing worms can *randomly* transfer a part of the BGP routing prefixes to victims until the payload is small enough (how small the payload should be can be arbitrarily

[4]In reality, because of congestion caused by the worm, only the beginning part of Slammer's propagation follows simple epidemic model (1) [6]. Thus for accuracy, we should use the two-factor worm model presented in [15] to model Slammer. Here we still use simple epidemic model for the purpose of illustration.

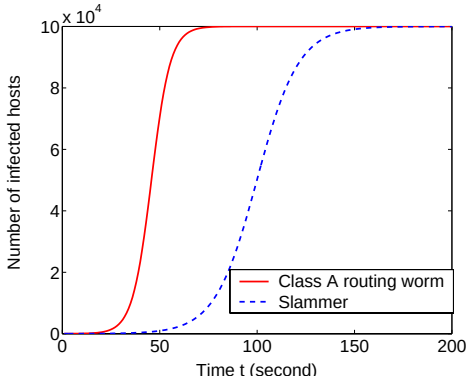


Fig. 5. Worm propagation comparison between a routing worm with Slammer (note that the time scale here is "second" instead of "minute" as in previous figures)

determined by attackers). In this way, after several rounds of infection, those newly infected hosts will have smaller payloads to transfer and also smaller IP spaces to scan. In this case, BGP routing worms can propagate in a "divide and conquer" approach. Because different infected hosts will have different sets of BGP prefixes, most BGP prefixes are likely to be preserved in some infected hosts, thus most vulnerable hosts on the Internet can still be scanned and infected by the BGP routing worms.

A similar "divide and conquer" approach has been mentioned in [11] to divide the hit-list for a hit-list worm among infected hosts. But its authors did not explain how such approach affected a hit-list worm's propagation. In the case of BGP routing worms, the "divide and conquer" approach has its advantages and disadvantages. The advantage is that the BGP prefix payload will be small after several rounds of partition, thus the worms are less likely to cause congestion to the Internet, and also it takes less time to infect a vulnerable host once the target is found. The disadvantage of this approach is that any IP address will be out of the scanning radar of some infected hosts that do not contain this IP address in their BGP prefixes. It is unclear how such approach can affect routing worms' propagation.

We attempt to understand the affectiveness of the divide and conquer approach through a simple analysis. Assume that when a routing worm infects a target, it passes half of its scanning space to the target (the space passed to the target includes the target host), then continues to scan the remaining half scanning space. We make the following assumptions: the set of N vulnerable hosts are uniformly distributed in the original scanning space Ω; no infected host will be removed; each worm copy uniformly scans IP addresses in its scanning space; during scanning, a worm copy independently chooses an IP address to scan, which means that it could possibly scan the same IP address repeatedly; initially there is only one infected host in the system.

For such a worm's propagation, when a host is infected and begins to scan and infect others, it is the only infected host in its scanning space. When there are overall I_t infected hosts, on average each infected host will have a scanning space $\Omega/I_t - 1$ (the host will not scan itself) and $N/I_t - 1$ vulnerable hosts in its scanning space. Therefore, the hitting probability of an infected host is

$$p = \frac{N/I_t - 1}{\Omega/I_t - 1} = \frac{N - I_t}{\Omega - I_t} \tag{7}$$

which decreases as I_t increases ($\Omega \gg N \geq I_t$).

Therefore, for a small time interval δ, the worm's propagation follows $I_{t+\delta} = I_t + \delta\eta \cdot p \cdot I_t$. The propagation model is:

$$\frac{dI_t}{dt} = \eta I_t \frac{N - I_t}{\Omega - I_t} \tag{8}$$

For Internet routing worms, $\Omega > 2^{30}$ while the number of vulnerable hosts N is usually much smaller than Ω. Therefore, $\Omega - I_t \simeq \Omega$. From (8), (4), and (3), we can derive

$$\frac{dI_t}{dt} \simeq \frac{\eta}{\Omega} \cdot I_t(N - I_t) = \beta I_t(N - I_t) \tag{9}$$

which is exactly the simple epidemic model (1). It means that under the simplified situation, a worm using the "divide and conquer" approach has the same propagation speed as an uniformly scanning routing worm, but with a much smaller BGP prefix payload to transfer.

Note that if a routing worm has the same order of propagation speed as Code Red or Blaster, then transferring of BGP prefix payload will not cause too much trouble to the worm. For example, for the BGP routing worm shown in Fig. 4(a)(b), on average the scans sent out by a worm copy hit anyone among the population N once in every 9.5 minutes [5]. In other words, a routing worm copy only needs to transfer the BGP prefixes payload on average once in every 9.5 minutes. If routing worms have such a propagation speed, then attackers probably do not need to worry about reinfection, and thus can make their programming tasks simpler.

VII. Defense Against Routing Worms

Routing worms use very simple ideas and publicly available information. The primitive forms of the central ideas of routing worms — reducing scanning space and selective attack — have already been implemented by attackers in Code Red II [32]. In this section we focus on how to defend routing worms quickly before attackers implement one of them.

Routing worms use public available information that are difficult or impossible to hide from attackers. IANA Class A address allocation has to be public. BGP routing tables are useful for the development of the Internet and have been used in various researches for many years; making BGP routing tables confidential may damage Internet developments. Even if BGP routing tables become confidential, there are hundreds to thousands EBGP routers on the Internet around the world; attackers can compromise any one of them first to get near-perfect information of global BGP routing prefixes. Many papers have discussed how to derive geographic information about IP addresses. Thus attackers can use those tools or methods to find out the geographic information of BGP prefixes by themselves. In addition, BGP routing prefixes, especially the mappings from prefix to AS, from AS to country, and IANA Class A address allocation, are all long-time stable information. Even if we make them not public available to attackers, they can simply use one-year-old data in their routing worms.

Furthermore, routing worms are very easy for attackers to implement, easier than the hit-list worm [11]. For a hit-list worm, attackers need to do some works to collect a hit-list of vulnerable computers, which is not so easy for some worms such as Slammer (the vulnerable hosts for Slammer, Windows SQL servers, do not advertise their addresses).

A. Effective defense: upgrading IPv4 to IPv6

For the reasons above, we cannot prevent the appearance of routing worms. Fig. 4 shows how much faster a routing worm can propagate when the worm reduces its scanning space by half or two thirds. Thus if we use the same principle to dramatically increase a scan-based worm's scanning space, we can prevent it from spreading out. Therefore, we believe that an effective defense against routing worms or any other scan-based worm is to upgrade the current IPv4 Internet to IPv6 — the vast address space of IPv6 (2^{64} IP addresses for a single subnetwork) can prevent a worm from spreading through scanning.

[5]The hitting probability of each scan is $p = N/(28.6\% \cdot 2^{32})$. The event of a worm's scans hitting a target is a bernoulli trail. Thus the average time used before a hitting is $1/(p\eta) = 9.5$ minutes.

IPv6 was designed as a replacement for IPv4. IPv6 has dramatically increased IP space from 32-bit IPv4 address to 128-bit addresses, which means that IPv6 has 2^{128} addresses instead of current 2^{32} IP addresses. Because of this huge IP address space, IPv6 implements hierarchical addressing theme. Fig. 6 shows the addressing format of the IPv6 global unicast addresses [19][20].

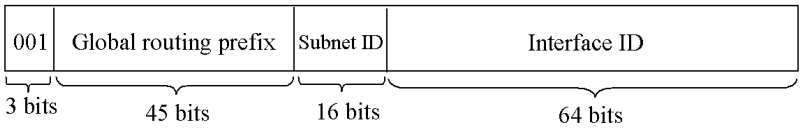


Fig. 6. IPv6 unicast address format

The first 3 bits of IPv6 unicast addresses are "001", which means that IANA has allocated 1/8th of the IPv6 space to unicast. A site has 80-bit IP addresses, among which a site can use 16 bits for subnetwork routing, and 64 bits for each subnetwork. Therefore, the smallest network in IPv6 has 2^{64} IP addresses (with prefix /64). In other words, the smallest network in IPv6 contains the number of IP addresses equal to that of *4 billion* IPv4 Internets.

Some people might think that allocating such big address space for a single subnetwork wastes too many IP resources. Actually, it does not. As shown in Fig. 6, 45 bits are used for global routing. Suppose there are *100 billion* people on earth, then on average each person can own 350 site networks (with prefix /48) — each of these site networks consists 65536 (2^{16}) the smallest subnetworks mentioned above.

Because of multihoming, many prefixes in the current IPv4 BGP routing tables overlap with each other, messing up the hierarchical structure of IP prefixes. In IPv6, A multihomed network will have multiple IP prefixes — when a network requests an additional path through another ISP, the new ISP simply allocates another prefix for the network [24]. IPv6 can also maintain its hierarchical addressing structure through "renumbering" and "autoconfiguration" [24], which enable dynamic changing of IP addresses and "plug and play" similar to current Dynamic Host Configuration Protocol (DHCP).

For IPv6, the BGP routing table contains the global routing prefixes (/3$\sim$/48). Some ISPs also directly allocate /64 prefixes, thus attackers can get information about BGP routing prefixes from /3 to /64. Inside a subnetwork, the address allocation is confidential and known only to the administrators of the network, thus attackers are unlikely to know the IP address allocation within a subnetwork containing 2^{64} IP addresses. For a particular subnetwork, attackers might know address information by hacking into one internal computer that happens to have the information, but attackers cannot know such information for all subnetworks on the Internet.

Even one single subnetwork in IPv6 will have sufficient IP space to defeat scan-based worms. Suppose there are $N = 1,000,000$ vulnerable hosts in one single subnetwork and a worm has scan rate $\eta = 100,000$ per second with 1000 initially infected hosts. Suppose the worm only scans and infects hosts in this subnetwork, thus $\Omega = 2^{64}$. Based on (4) and the epidemic model solution (2), when the worm infects I_T hosts at time T, the time T will be

$$T = -\frac{\Omega}{\eta N} \cdot ln[\frac{I_0 N / I_T - I_0}{N - I_0}] \tag{10}$$

From this equation, we know that if such a worm wants to infect 50% of vulnerable hosts in this single subnetwork, the worm will need to spend *40 years* to achieve that goal.

B. IPv6 recommendations for defending scan-based worms' attacks

In order to defend against routing worms, or any other scan-based worms, simply upgrading IPv4 to IPv6 is not enough: the IP address allocation in an IPv6 subnetwork should be random. Any address

assignment method that is not random can facilitate a scan-based worm attack as long as attackers know the assignment method. For example, originally the 64-bit address in a subnetwork is determined by the network interface embedded IEEE identifier — a link-layer MAC address [21]. Like Internet addresses, the MAC addresses are assigned by IEEE to various network-equipment manufacturers. Thus attackers can know MAC address allocation, too. For this reason, and also for the privacy concern about MAC addresses, administrators should assign random addresses inside a local subnetwork.

In IPv6 Internet, attackers can still use information from the BGP routing tables, thus the IP address space contained in each BGP prefix should not be too small. When allocating IP prefixes, Internet Registries and ISPs should strictly obey the hierarchical structure (or similar structure) of IPv6 as shown in Fig. 6. In this way, attackers cannot know address allocation for networks smaller than a /64 prefix network.

Of course, upgrading IPv4 to IPv6 is not the omnipotent solution for defending worms' attacks. The vast IP space of IPv6 is useful to prevent scan-based worms, such as Code Red, Code Red II, Slammer, and Blaster. For Nimda, IPv6 can prevent its spread through scanning, but not its spread through email and other ways. The IP space of IPv6 does not help in defending against topological worms, such as the Morris worm [10], or mass-mailing email viruses, such as "SoBig" series [27]. Fortunately, non-scan-based worms, such as topological worms or metaserver worms [14], are difficult to program; mass-mailing email viruses propagate much slower than recent worms. In recent years, most wide-spread worms belong to scan-based worms. Therefore, for the security reason of scan-based worms and many other reasons, such as IPSec, mobile networking, Peer-to-Peer Networking [18], it is better for us to make the IPv6 transition earlier.

VIII. Conclusions

In recent years, scan-based worms have become the major security problems for Internet and cost a lot damages to our society. In order to defend against future worms, we need to anticipate the next move of attackers. In this paper, we present a new theoretical scan-based worm called "routing worm", which can use information provided by Internet allocated addresses or BGP routing table to increase its propagation speed by two times to more than three times. The routing worm can also use the geographic information associated with addresses to conduct selective attacks to a specific target. Routing worms can be easily implemented by attackers and they could cause considerable damage to our Internet. Since routing worms are scan-based worms, we believe that an effective way to defend against routing worms and all other scan-based worms is to upgrade IPv4 to IPv6 — the vast address space of IPv6 can prevent a worm from spreading through scanning. At the end of this paper, we give a brief introduction of IPv6, hoping that it can help people to understand IPv6 better and help clearing some prejudices towards IPv6.

References

[1] Z. Chen, L. Gao, and K. Kwiat. Modeling the Spread of Active Worms, In *IEEE INFOCOM*, 2003.

[2] D.J. Daley and J. Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, 1999.

[3] J. O. Kephart and S. R. White. Directed-graph Epidemiological Models of Computer Viruses. In *Proceedings of the IEEE Symposimum on Security and Privacy*, 343-359, 1991.

[4] J. O. Kephart, D. M. Chess and S. R. White. Computers and Epidemiology. In *IEEE Spectrum*, 1993.

[5] J. O. Kephart and S. R. White. Measuring and Modeling Computer Virus Prevalence. In *Proceedings of the IEEE Symposimum on Security and Privacy*, 1993.

[6] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Security and Privacy Magazine*, 1(4):33-39, July 2003.

[7] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an Internet Worm. In *Proc. ACM/USENIX Internet Measurement Workshop*, France, November, 2002.

[8] David Moore, Ram Periakaruppan, Jim Donohoe, k claffy. Where in the World is netgeo.caida.org? in *INET 2000 Proceedings*, June 2000.

[9] C. Nachenberg. The Evolving Virus Threat. In *23rd NISSC Proceedings*, Baltimore, Maryland, 2000.

[10] D. Seeley. A tour of the worm. In *Proc. of the Winter Usenix Conference*, San Diego, CA, 1989.

[11] S. Staniford, V. Paxson and N. Weaver. How to Own the Internet in Your Spare Time. In *11th Usenix Security Symposium*, San Francisco, August, 2002.
[12] Venkata N. Padmanabhan, Lakshminarayanan Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM'01*, August 2001.
[13] Michael H. Warfield. Security Implications of IPv6. Internet Security Systems, Inc. White Paper. 2003.
[14] Nicholas Weaver, Vern Paxson, Stuart Staniford, and Robert Cunningham. A Taxonomy of Computer Worms. In *ACM CCS Workshop on Rapid Mal-code (WORM'03)*, Oct. 27, 2003.
[15] Cliff C. Zou, Weibo Gong, and Don Towsley. Code Red Worm Propagation Modeling and Analysis. In *9th ACM Conference on Computer and Communication Security (CCS'03)*, Washington DC, 2002.
[16] Cliff C. Zou, Lixin Gao, Weibo Gong, and Don Towsley. Monitoring and Early Warning for Internet Worms. In *10th ACM Conference on Computer and Communication Security (CCS'03)*, Oct. 27-31, Washington DC, USA, 2003.
[17] Hans-Werner Braun. BGP-system usage of 32 bit Internet address space. November 1997. http://moat.nlanr.net/IPaddrocc/
[18] Tim Chown, Jeff Doyle, Gary Hemminger *et al.* IPv6: An Internet Evolution. http://www.ipv6forum.org/navbar/papers/IPv6-an-Internet-Evolution.pdf
[19] R. Hinden, S. Deering. RFC-3513: Internet Protocol Version 6 (IPv6) Addressing Architecture. April 2003.
[20] R. Hinden, S. Deering, E. Nordmark. RFC-3587: IPv6 Global Unicast Address Format. August 2003.
[21] R. Hinden, S. Deering. RFC-2373: IP Version 6 Addressing Architecture. July 1998.
[22] CAIDA. Dynamic Graphs of the Nimda worm. http://www.caida.org/dynamic/analysis/security/nimda/
[23] T. Narten, R. Draves. RFC-3041: Privacy Extensions for Stateless Address Autoconfiguration in IPv6. January 2001.
[24] Florent Parent. IPv6 tutorial. http://www.viagenie.qc.ca/en/ipv6/presentations/ripe40-ipv6tutorial-praha-oct2001.pdf
[25] Uri Raz. How to Find a Host's Geographic Location. http://www.private.org.il/IP2geo.html
[26] Nicholas C. Weaver. Warhol Worms: The Potential for Very Fast Internet Plagues. http://www.cs.berkeley.edu/ nweaver/warhol.html
[27] CERT Incident Note IN-2003-03: W32/Sobig.F Worm. http://www.cert.org/incident_notes/IN-2003-03.html
[28] CAIDA. IP v4 Address Space Utilization. 1998. http://www.caida.org/outreach/resources/learn/ipv4space/
[29] Cisco Documents. Border Gateway Protocol (BGP). http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/bgp.htm
[30] University of Oregon Route Views Project. http://www.routeviews.org/
[31] eEye Digital Security. Blaster Worm Analysis. http://www.eeye.com/html/Research/Advisories/AL20030811.html
[32] eEye Digital Security. CodeRedII Worm Analysis. http://www.eeye.com/html/Research/Advisories/AL20010804.html
[33] CAIDA. IPv4 BGP Geopolitical Analysis. http://www.caida.org/analysis/geopolitical/bgp2country/
[34] CAIDA. Visualizing Internet Topology at a Macroscopic Scale. http://www.caida.org/analysis/topology/as_core_network/
[35] Net World Map project: IP Address Locator Tool. http://www.geobytes.com/IpLocator.htm?GetLocation
[36] Route Views Project. http://www.routeviews.org/
[37] RIPE NCC Routing Information Service. http://www.ripe.net/ris/
[38] CERNET BGP VIEW Global Internet. http://bgpview.6test.edu.cn/bgp-view/index.shtml
[39] Reserved IPv4 addresses. http://www.cidr-report.org/v6/reserved-ipv4.html
[40] IANA IPv4 Address Registry - Current Issues. http://www.cidr-report.org/images/iana-v4.html
[41] Akamai Service: EdgeScape. http://www.akamai.com/en/html/services/edgescape.html
[42] USA Today News. The cost of Code Red: $1.2 billion. http://www.usatoday.com/tech/news/2001-08-01-code-red-costs.htm
[43] CNN News. Computer worm grounds flights, blocks ATMs. http://europe.cnn.com/2003/TECH/internet/01/25/internet.attack/

Appendix

There are 42 Class A addresses have been assigned to companies or organizations according to IANA's latest update [39]. However, 14 Class A are inactive and we cannot find them in the BGP routing table of September 2003. These 14 Class A addresses are:

Prefix	Registry purpose
003/8	General Electric Company
009/8	IBM
011/8	DoD Intel Information Systems
019/8	Ford Motor Company
021/8	DDN-RVN
022/8	Defense Information Systems Agency
026/8	Defense Information Systems Agency
028/8	DSI-North
029/8	Defense Information Systems Agency
030/8	Defense Information Systems Agency
046/8	Bolt Beranek and Newman Inc.
048/8	Prudential Securities Inc.
051/8	Department of Social Security of UK
054/8	Merck and Co., Inc.