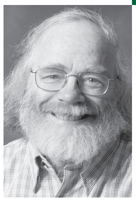
STEVEN M. BELLOVIN, BILL CHESWICK, AND ANGELOS D. KEROMYTIS

worm propagation strategies in an IPv6 Internet



Steve Bellovin, a member of the National Academy of Engineering, is a professor of Computer Science at Columbia University. He is one of the creators of Netnews, a long-time researcher on network security, and co-author of the first book on firewalls.

smb@cs.columbia.edu

Bill Cheswick is Chief Scientist at Lumeta, a company he co-founded to explore commercial intranets and the Internet. Ches did early work on firewalls and proxies and is co-author of the first book on firewalls.

ches@cheswick.com

Angelos Keromytis received his Ph.D. in computer science from the University of Pennsylvania in 2001. Currently he is an associate professor of computer science at Columbia University. His research interests include self-healing software, system reliability, design and analysis of network and cryptographic protocols, and denial-of-service protection.

angelos@cs.columbia.edu

IN RECENT YEARS, THE INTERNET HAS been plagued by a number of worms. One popular mechanism that worms use to detect vulnerable targets is random IP address-space probing. This is feasible in the current Internet due to the use of 32-bit addresses, which allow fast-operating worms to scan the entire address space in a matter of a few hours. The question has arisen whether or not their spread will be affected by the deployment of IPv6. In particular, it has been suggested that the 128-bit IPv6 address space (relative to the current 32-bit IPv4 address space) will make life harder for the worm writers: assuming that the total number of hosts on the Internet does not suddenly increase by a similar factor, the work factor for finding a target in an IPv6 Internet will increase by approximately 2^{96}, rendering random scanning seemingly prohibitively expensive.

Some worms, such as Melissa, spread by email. These worms will not be affected by the adoption of IPv6; though the space of possible email addresses is vast, these worms typically consult databases such as Microsoft Outlook's address book.

On the other hand, life will indeed be harder for address-space scanners, such as Code Red and Slammer. Clever heuristics can cut the search space dramatically. More specifically, multi-level searching and spreading techniques can negate the defender's advantage. However, the code size required for worms will increase, which may help prevent Slammer-like attacks. This has created the impression that an IPv6 Internet would be impervious to similar kinds of worms.

In the past, there have been two forms of address-space scans. Some worms use a uniformly distributed random number generator to select new target addresses. This strategy is indeed unlikely to succeed in an IPv6 world. Other worms preferentially spread locally, by biasing the search space toward addresses within the same network or subnet. This will be a more successful strategy, though at first glance the 80-bit local space (nearly twice Avogadro's number!) would seem to be a formidable obstacle. We observe that certain strategies can improve the attacker's odds. In particular, by taking advantage of local knowledge and patterns in address-space assignment, the attack program can cut the search space considerably.

We discuss a number of strategies worms could use in an IPv6-based Internet to find new targets. We separate these into two categories, wide-area and local-area searches, somewhat mirroring the IPv6 address architecture. We argue that worms will use different types of information sources to first determine existing networks and establish a presence there, and then spread locally inside an organization. We hope to illustrate that simple reliance on the IPv6 address space for protection against scanning worms is not a wise defensive strategy, and we suggest areas where research could assist in detecting and limiting future worm propagation.

The IPv6 Addressing Architecture

Addresses in the IPv6 addressing architecture, defined in RFC 3513, come in a number of different flavors. Those of interest to us are link-local addresses, unique local addresses, global addresses, and multicast addresses.

All forms of unicast address are conceptually divided into two pieces, a network section and a host section. Roughly speaking, the network section identifies the particular LAN; the host section identifies the particular node on the LAN. In fact, both sections have internal structure. Furthermore, the address is generally divided into two 64-bit halves. (There are subtleties that lie outside the scope of this article.)

In the network section of the address, the first 10 bits denote the scope of the address. The next 38 bits identify the site and (implicitly) the ISP, as explained in RFC 3177. In order to promote hierarchical addressing, only the largest ISPs have their own address allocations; smaller ISPs are assigned space by their upstream provider. Identifying the set of all ISPs considerably reduces the search space for the attacker.

The next 16 bits in the network section of the address identify the subnet within each site. No site will have 2^{16} subnets, though identifying the allocated subnets could pose a challenge for the attacker.

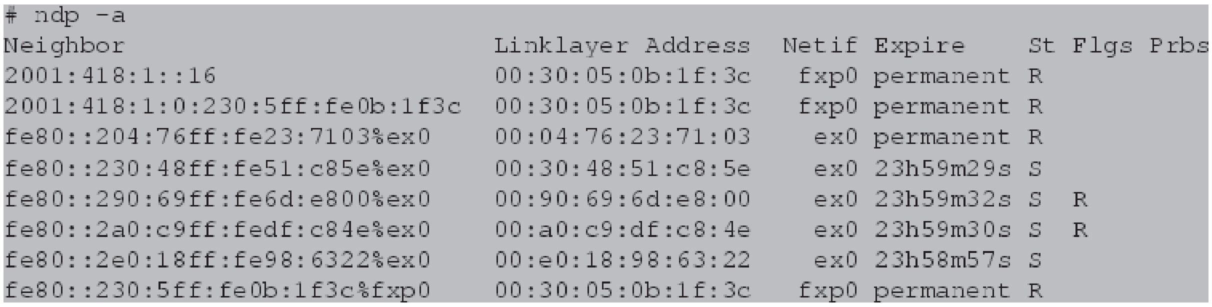
There are several possible formats for the host identifier (the last 64 bits of the address). Clients will often generate their own addresses via stateless auto-configuration, as described in RFC 2460.

Sources of Information

As we already mentioned, IPv6 worms can spread by using a two-level strategy. Here we present several information sources, divided into local and wide-area sections. We do not claim that this list is exhaustive; however, the list we do present is probably sufficient and is undoubtedly indicative of a much larger class of information sources that could be exploited.

LOCAL INFORMATION SOURCES

NEIGHBOR DISCOVERY



```
# ndp -a
Neighbor                          Linklayer Address  Netif Expire     St Flgs Prbs
2001:418:1::16                    00:30:05:0b:1f:3c   fxp0 permanent R
2001:418:1:0:230:5ff:fe0b:1f3c    00:30:05:0b:1f:3c   fxp0 permanent R
fe80::204:76ff:fe23:7103%ex0      00:04:76:23:71:03    ex0 permanent R
fe80::230:48ff:fe51:c85e%ex0      00:30:48:51:c8:5e    ex0 23h59m29s S
fe80::290:69ff:fe6d:e800%ex0      00:90:69:6d:e8:00    ex0 23h59m32s S  R
fe80::2a0:c9ff:fedf:c84e%ex0      00:a0:c9:df:c8:4e    ex0 23h59m30s S  R
fe80::2e0:18ff:fe98:6322%ex0      00:e0:18:98:63:22    ex0 23h58m57s S
fe80::230:5ff:fe0b:1f3c%fxp0      00:30:05:0b:1f:3c   fxp0 permanent R
```

FIGURE 1: NEIGHBOR DISCOVERY TABLES ON AN IPV6 HOST.

In IPv6, Neighbor Discovery as described in RFC 2461 is used to map IP addresses to local network addresses (e.g., Ethernet addresses), similar to the way ARP is used in today's IPv4 networks. As such, it can be a rich source of information about machines on the LAN. A sample listing of a Neighbor Discovery table is shown in Figure 1. A worm that has infected a node in the LAN can thus determine the addresses of other existing nodes in the same LAN.

ROUTING TABLES AND PROTOCOLS

Typically, host routing tables only contain entries for other local hosts plus a default route entry for all traffic outside the LAN. Many organizations, however, internally run routing protocols such as OSPF or RIP. The few IPv6 networks we are familiar with actually use RIPng (an adaptation of RIP for IPv6 networks, described in RFC 2080), and in the future may run OSPFv6 or IS-IS. In such an environment, a worm would be able to either directly consult the host routing tables (e.g., using the UNIX netstat command) or participate in a routing protocol, if only as a passive listener. In either case, the worm would be able to determine other valid subnets within the organization and subsequently target those [1].

INTERFACE IDENTIFIERS

The Neighbor Discovery tables provide another useful hint: a list of some locally used network cards. If stateless autoconfiguration is used, the high-order 32 bits of the low-order (host section) 64 bits of the IPv6 address identify the manufacturer of the card. In many organizations, common purchasing patterns mean that LAN cards in use will largely be from a small set of manufacturers. (We informally sampled two large, heterogeneous LANs, one educational and one corporate. In each case, there was a reduction to about 40 different card types, from 161 and 227 hosts, respectively.) For each such manufacturer identifier, there are at most 2^{24} possible addresses. This is a search space comparable to what is successfully exploited by today's IPv4 worms.

MULTICAST PING

Multicast is a fundamental part of IPv6 design, which unfortunately can be abused for target discovery by worms. RFC 3513 notes that FF0E:0:0:0:0:0:0:101 addresses "all NTP servers in the Internet." An NTP query to that address might locate many victims. While such a packet is unlikely to traverse the entire Internet, FF05:0:0:0:0:0:0:2 would find all routers at a site, and FF02:0:0:0:0:0:0:1 would send to all hosts on the local link. Fortunately, FF05:0:0:0:0:0:0:1, the larger-scope analog to send to all hosts at a site, is not defined.

A related IPv6 concept is that of "anycast" addresses, defined in RFC 2526, which can be used to locate the "closest" instance of a service. A worm can exploit this and other service-location mechanisms such as SLP, DHCP, DNS, and LDAP to locate local targets for attack. Service-location mechanisms are likely to be increasingly used in an IPv6 Internet, both because of increased host mobility and due to the difficult-to-memorize addresses.

HOST CONFIGURATION AND LOG FILES

Computers are generally configured with the addresses of other important local computers, such as email gateways, local file servers, Web proxy servers, local DNS servers, the /etc/hosts file in UNIX, SSH known_hosts files, *etc*. A sufficiently versatile worm could examine likely places for such configuration files—the registry on Windows machines is one such location—to discover other attack targets. Furthermore, although a worm may use a non-email infection vector (e.g., a buffer overflow for a popular service), it can still use archived user email to find new targets (hostnames).

DNS ZONE TRANSFERS

Typical DNS servers are configured such that they do not allow zone transfers from hosts other than the authorized secondary servers. However, some organizations have a mixed record on restricting zone transfers from hosts inside that organization. Thus, it may be possible for a worm to acquire a complete list of all hosts in a domain, once a host inside that domain has been infected; this list would include all hosts with static addresses as well as those using Dynamic DNS Updates.

PASSIVE EAVESDROPPING

Although most local area networks are switched, wireless networks offer the potential for discovering new targets on the local network simply by monitoring traffic. Furthermore, random-address flooding can be used in networks such as Ethernet to force a switch to effectively broadcast all local traffic and incoming external traffic.

WIDE-AREA INFORMATION SOURCES

Wide-area information sources can be used to determine valid IPv6 prefixes (networks) to target. Often, they also provide the addresses of valid hosts (typically servers) in that domain. Even when they do not, however, they can be used as a starting point for scanning. Although we do not have sufficient data, an informal poll of network operators suggested that servers would be assigned addresses statically, and that these addresses would be located in the low end of the subnet address range, significantly easing the task of a scanning worm.

ROUTING PROTOCOLS

Routing protocols provide information on address prefixes that are in use. These can be used both locally and across the Internet.

Local use is easy: the attack program just listens to local routing traffic. This may require joining the "all routers" multicast group, but there are no access controls that would prevent that from happening.

Remote use is more interesting, but perhaps more problematic. There are no inherent IPv6 features that would permit easy capture of BGP routing information by an ordinary host. On the other hand, there are public archives of routing data, such as the one available at www.routeviews.org. If this data is available for IPv6—and it is a valuable operational and research resource—a worm could use it for propagation purposes.

SERVER LOGS

Web, DNS, and incoming email servers are typically contacted by client machines from many different places. The log files of such contacts offer a good mechanism for wide-area spread. A more ambitious worm could kill off the legitimate server and grab its port number, thus collecting new addresses in real time.

SERVER ADDRESSES

Anecdotal evidence suggests that IPv6 servers tend to have low-numbered addresses. The prefix alone is hard to remember; administrators tend to select easily memorizable values for the low-order bits. This human tendency can be exploited by worms.

SUBVERTING NEIGHBOR DISCOVERY

A worm-infected host could impersonate the LAN router using Neighbor Discovery and divert all traffic to/from external hosts to itself. Such attacks are known and exploited in the current IPv4 Internet (e.g., the dsniff toolkit); while they are more difficult in an IPv6 environment, they are still possible. Using this attack, a worm would be able to find valid IPv6 addresses outside the local area network (whether in remote organizations or other LANs within the same organization). Passive eavesdropping can be equally fruitful in determining remote IP addresses (by capturing incoming packets), as discussed previously.

SEARCH ENGINES AND DNS

Web search engines are a particularly attractive source of information on potential targets, especially if the worm is targeting Web servers, as was the case with Code Red. Although such engines typically only point to Web servers, they can be used to identify valid prefixes by determining the hostname of a Web server and resolving its IPv6 address through DNS. Likewise, DNS itself can be used as a search engine for valid hostnames, by exhaustively searching for all words (and combinations of words) from a dictionary. In [2], we showed that a DNS worm in IPv6 could spread as fast as an IPv4 address-scanning worm.

PEER-TO-PEER PROTOCOLS

The most intriguing form of wide-area data is peer-to-peer networks. By participating in topology maintenance, watching queries and responses, and sending out occasional queries of its own, a worm could learn the addresses of many different hosts. File-swapping networks such as Morpheus, Kazaa, and Gnutella offer particularly attractive targets, as do more "traditional" presence protocols such as IRC, Jabber, and others.

Strategies for Spreading

Based on our discussion of information sources in the previous section, we believe that scanning worms in an IPv6 Internet will use a two-phase approach for discovering and infecting targets:

- Discover valid IPv6 prefixes using search engines, server logs, routing table information, etc. These sources may indicate specific targets within those prefixes (e.g., a Web server listed on a search result, or a host participating in a peer-to-peer network), or simply the valid prefix (as may be the case with getting a copy of a BGP table). In the second case, targeted address scanning may be needed, but by starting at the low end of the range a worm will maximize its chances of finding a server.
- Once inside an organization's network, use local information sources to determine the identity (address or hostname) of other nodes to infect. The repertoire of the worm is significantly richer here, and we believe that vulnerable nodes will be infected fairly quickly once the worm has established a presence.

This two-phase approach can also be extended to propagation. Intuitively, propagation across organizations calls for an approach distinct from that used for spreading within organizations. We believe that multi-partite worms such as Nimda will appear more frequently: email or Web-downloadable executable content (e.g., Java or Javascript embedded in every page served by a Web server) is particularly useful in propagating across administrative domains, as it appears to be difficult to intercept at the firewall [3]. A worm that manages to infect a popular Web server will be able to propagate widely and quickly to many different networks, potentially without raising suspicion for some time (pull model); email worms (push model) can exploit the social and professional interactions between individuals and organizations to spread.

More generally, client/server worms can operate efficiently in two different modes. In client mode, they search for and infect servers of some type. Once they've penetrated a server, they use a different technique to attack clients that connect to it.

Once a worm has managed to penetrate a new environment, it can switch to something more akin to traditional address scanning, using the information gathered using the techniques described in "Local Information Sources," above, as hints to direct the scanning process.

Discussion

The problem of locating hosts is not limited to the authors of malware. Network administrators and security officers responsible for intranets have a keen interest in the population of hosts found on their networks. They generally have extensive tools for auditing and updating such hosts to keep them up to date. Network management companies are often paid according to the number of hosts they manage. And, of course, unknown and unregistered hosts that appear on an intranet can be a concern, possibly violating perimeter security or network connection policy.

On traditional IPv4 intranets, the various techniques described above, along with simple or multi-protocol network probes, are used for host discovery. A computer inventory, especially including MAC address information, can be quite useful for tracking hosts. On IPv4 networks, MAC information is obtained, via SNMP, of ARP caches in routers. This incomplete information is about the best we can do.

In principle, the Ethernet addresses can be monitored as part of the IPv6 addresses as traffic travels through company checkpoints: the bottom 48 bits of an IPv6 address are supposed to be the MAC address. Whether this is the MAC address or a small integer, as we have seen, or even a cryptographically derived address as proposed in RFC 3972, well-placed flow monitors can collect census information. Similar information is already available from routers on a read-only basis using SNMP version 1 or 2, which has a sniffable community string. SNMPv3 is not widely used, but should be—as we have seen, network population information is going to become more sensitive.

A census of local IPv6 addresses could be kept in each router, up to a point. These could be collected and consolidated by authorized network administrators, but should be protected better than current router contents. Network discovery would proceed in two stages: first, discover the routers, perhaps with traceroute-style Internet mapping techniques, then gain administrative access to the router and dump the flow history information.

In any case, network administrators will be in the same game as the virus and worm writers, but with the home-field advantage. They need new tools for IPv6 networks to collect this data, with better protection of the acquired data from access by malware.

Conclusion

We have outlined a number of techniques that scanning worms can use in an IPv6 Internet to locate potential targets. These techniques are equally applicable to the current IPv4 Internet, albeit not as efficient as random scanning. Although "conventional" address-space scanning is prohibitively expensive in that environment, we believe that the diversity of sources we discussed (which is by no means exhaustive) guarantees a rich target set for worms.

The implication is that we cannot afford to rest on the assumption of inherent protection in the IPv6 addressing scheme; further research in worm detection and containment is needed. For our future work, we plan to investigate how much "coverage" our techniques can give us in the current Internet (as a measure of the effectiveness of the approach), as well as determine ways of monitoring requests to these information sources that could reveal worm-scanning activity.

REFERENCES

[1] C.C. Zou, D. Towsley, W. Gong, and S. Cai, "Routing Worm: A Fast Selective Attack Worm Based on IP Address," in *Proceedings of the Workshop on Principles of Advanced and Distributed Simulation (PADS)*, June 2005.

[2] A. Kamra, H. Feng, V. Misra, and A.D. Keromytis, "The Effect of DNS Delays on Worm Propagation in an IPv6 Internet," in *Proceedings of IEEE INFOCOM*, March 2005.

[3] D.M. Martin, S. Rajagopalan, and A.D. Rubin, "Blocking Java Applets at the Firewall," in *Proceedings of the Symposium on Network and Distributed System Security*, February 1997.