IPv6 Host Address Usage Survey

Qinwen Hu and Nevil Brownlee

***Abstract*—It is tempting to assume that for IPv6, with its 64-bit Interface IDs (IIDs), some existing address scanning attacks have become infeasible. RFC 5157 suggests how Interface IDs could be allocated so as to minimize a site's vulnerability to address scans, essentially by using IIDs consisting of a pseudorandom sequence of 1s and 0s. In this paper, we investigate how network administrators are actually allocating their Interface IDs. We have developed and carried out a survey of various IPv6 addresses from 50 countries. We find that few network administrators are using RFC 5157's allocation methods; instead we find that most network administrators are using one of five simple allocation schemes which tend to leave zero bits in large sections of their Interface IDs. We observe that such schemes can leave networks vulnerable to address scanning.**

***Index Terms*—Address allocation mechanisms, IPv6, privacy, security.**

I. Introduction

In the early 1990s it became clear that the 32-bit IPv4 space would eventually limit the Internet's growth; since then that small address space has led to many scanning attacks happening in the IPv4 network. When the IPv6 protocol was proposed, it extended the Internet address space to 128 bits. In [1], Chown mentioned that address scanning attacks may be less common in the IPv6 network because of its larger addresses. Before 2003, IPv6 traffic levels were low; over the last ten years, they have risen - for example, 6lab.cz [1] observed that all over the world only 25.6% of DNS servers have A and AAAA resource records in the DNS domain. By searching the existing IPv6 addresses for some large organizations, such as Google, Facebook and YouTube, we find that network administrators do not use the recommended methods from [1] to allocate their Interface ID fields. Searching for a better understanding of the existing Interface ID field allocation mechanisms, we designed a survey to collect IPv6 addresses from DNS servers and built a database to store those addresses into well-defined subsets; more detailed descriptions are given in Section III. Our survey not only yielded an overview of existing IPv6 address allocation mechanisms, but also demonstrates that it is possible to launch an address-scan attack in existing IPv6 networks. We first review related work in the IPv6 survey and security fields, then summarize our research objectives in Section II. In Section III we describe the data and our methods. Section IV focuses on establishing baseline characteristics of IPv6 addresses and discusses the usage of IPv6 address allocation mechanisms in various countries. Finally, Section V summarizes our conclusions.

Manuscript received January 10, 2014; revised March 25, 2014.

Qinwen Hu is with at computer science department of the University of Auckland, Private Bag 92019, Auckland 1142, New Zealand (e-mail: qhu009@aucklanduni.ac.nz).

Nevil Brownlee is with the CAIDA (the Cooperative Association for Internet Data Analysis), the WAND Network Research group at the University of Waikato and the IPFIX (IP Flow Information Export) Working Group.

[1] http::/6lab.cz

II. Related Works

There is a growing momentum to deploy IPv6 so as to support the growth of new Internet technologies and users. Over the last decade, many IPv6 studies have explored IPv6 address allocation strategies and IPv6 security issues.

In [2], T. Aura proposed how that one could use a public key to generate the interface identifier (ID) portion for the IPv6 address. He calls these addresses Cryptographically Generated Addresses (CGAs). This solution suggests an authentication channel between sender and receiver, because the receiver can verify address ownership by checking the IPv6 address and CGA parameters. Moreover, CGA is a stateless auto-configuration solution; users can get the IPv6 address automatically without the need for manual user configuration. However, in [3], Alsadeh *et al.* have mentioned some weaknesses of using the CGA mechanism. For example, generating the CGA address may need a large amount of time and resources. Moreover, attackers can easily create a valid address from their own public keys or they can capture Neighbor Discovery messages and modify the sender's CGA parameters; in this scenario the CGA verification process on the receiver's side will fail.

In [4], Hinden *et al.*, described another auto-configuration approach which is called IEEE EUI-64. This mechanism eliminates the need for manual configuration or DHCP6 to assign the IPv6 addresses for a new host. The mechanism uses three fields. The first 24 bits of the ID field will use a network device's Organizationally Unique Identifier (OUI), followed by the 16 bits 0xFFFE. The remaining 24 bits of the MAC address will fill the lower 24 bits in the IID field. The main advantage of using this solution is that it needs no pre-configuration or additional security requirement. Any node can get an IPv6 address when it is connected to the network. Although EUI-64 is easy to use, there are some disadvantages to it. In [5], Cooper *et al.* emphasize a few weaknesses of using IEEE EUI-64 identifier [6], for example: attackers can observe user activities by monitoring IPv6 addresses. Moreover, the IPv6 address structure is divided between a topological portion and an interface identifier portion; if the interface identifier remains the same when a host moves to a different network, it is possible for attackers to track the movements of that host. Furthermore, they think the EUI-64 mechanism could cause device vulnerability exploitation, because it embeds the OUI information into the IID field.

In [7], Carpenter *et al*., mentioned that it is possible to

DOI: 10.7763/IJFCC.2014.V3.323

detect the live IPv6 hosts inside a subnet. They pointed out that if an attacker has access to the target subnet, then he could use Neighbor Discovery and ping6 to the link scope multicast address (FF02::1). If a node on the same link is listening to such a multicast query, it will reply to the query with its link-local address. After that, the attacker can apply the global prefix to that and use it to reach the active hosts in this subnet.

In [1] and [8], the authors discussed how it would be possible to remotely launch address scanning attacks. They pointed out that some existing transition solutions employ special mechanisms to allocate IPv6 addresses, using IPv4 addresses in the 64 bit Interface ID field. For example, if the host is using 6to4 to allocate an IPv6 address in the Microsoft environment, then the IPv6 address will look like 2002:V4ADDR::VADDR. In that environment, an attacker can reduce the 128-bit IPv6 address search space to a 32-bit IPv4 address space.

However, in the existing IPv6 network, some network administrators ignore the recommended IPv6 address allocation mechanisms, and use simpler mechanisms instead. We think that some potential security issues can be studied by surveying existing IPv6 address allocation mechanisms. After reviewing the related work in IPv6 surveying, we decided to extract IPv6 addresses from reverse DNS servers by using reverse DNS lookups. In [9] section 2.1, it is suggested that, "Every Internet reachable host should have a name ... Make sure your PTR and A records match. For every IP address, there should be a matching PTR record in the in-addr.arpa domain".

III. Methodology

An IPv6 address contains a Routing Prefix (48 bits), a Subnet number (16 bits) and an Interface IDentifier (64 bits). The IPv6 routing prefix can easily be found from a Regional Internet Registry (RIR) website or from some IPv6 research websites, e.g. IPv6 Deployment Status [10]. The existing address-scanning strategies can explore the 16-bit Subnet Id field easily. Therefore, [1] suggested the IPv6 Interface ID field should be filled by random bits; so as to help prevent IPv6 address scanning attacks in the existing IPv6 network.

Our objective is to investigate the IPv6 mechanisms currently used to allocate Interface IDs. Therefore, we set up a survey of the IPv6 environment. The survey involves two steps. First, we choose the top fifty countries which have the largest number of assigned ::/48 IPv6 address blocks. Then we launch our test program to probe each target address block and save its DNS responses. We give a detailed description in the following sections.

A. Data Set

The IPv6 network is not yet widely deployed, so it is challenging for us to decide the survey countries. We use the Regional Internet Registries Statistics website [10] to select our survey countries; this website updates the IPv6 deployment status regularly. In order to better understand the existing IPv6 Interface ID allocation mechanisms, we needed more sufficient data to analyze. Therefore, we selected the fifty countries that had the largest number of assigned: /48 IPv6 addresses blocks.

B. Background: Reverse DNS Lookup

The DNS is one of the most significant components of today's Internet; it provides the mapping between domain names and IP addresses. A DNS server resolves two types of queries: forward and reverse lookup [11]. A reverse lookup attempts to map an IPv4 or IPv6 address to a corresponding domain name. DNS uses a hierarchical tree structure to organize the mapping between domain names and IP addresses. When network administrators add a new domain record, they should align both forward and reverse DNS tree structures. For an IPv4 address, the prefix octets in reverse order are prepended to the second level domain suffix 'in-addr.arpa' and stored as one node in the tree structure. For example, adding a new domain name 'www.cs.auckland.ac.nz' and IP address '130.216.33.163' into the in-addr.arpa. domain, '163.33.216.130.in-addr.arpa' is logically below '33.216.130.in-addr.arpa', which is one level below '216.130.in-addr.arpa'. In RFC 3596 [12], Huitema et al explained that for an IPv6 address the reverse mapping uses the same principle of reversing the address. However, [12] also specified that, "An IPv6 address is represented as a name in the ip6.arpa domain by a sequence of nibbles separated by dots with the suffix '.ip6.arpa'. The sequence of nibbles is encoded in reverse order, i.e., the low-order nibble is encoded first, followed by the next-lowest-order nibble and so on. Each nibble is represented by a hexadecimal digit.

For example, the reverse lookup domain name corresponding to the address 4321:0:1:2:3:4:567:89ab would be b. a. 9. 8. 7. 6. 5. 0. 4. 0. 0. 0. 3. 0. 0. 0. 2. 0. 0. 0. 1. 0. 0. 0. 0.0 .0.0.1.2.3.4 .ip6.arpa."

C. Reverse Lookup Algorithm

As a starting point for our survey tool we chose 'thc-ipv6'[2] which implements the reverse lookup search mechanism described in section B. thc-ipv6 tool uses a block of addresses whose size can be divided by 4 as an input parameter, such as /32, /48 and /64. The following sentences and Fig. 1 describe the search algorithm used by this tool.

DNS servers will send a different response for each request, depending on the records in DNS domains. In general, there are three common responses:

- 'NXERROR' means this '*.ip6.arpa' domain exists in the ip6.arpa domain, but there are no PTR records for it, When the program receives this message, it adds a new nibble and appends it to the previous reverse query. The initial value of the new nibble is 0.
- 'NXDOMAIN' means there are no records for '.*ip6.arpa' in the domain name space. The program will increase the the value on the current nibble and send the request again.
- If the response is the hostname, the program will save that hostname into our database.

For example, if an input address prefix is "2001:620:0::/48", Fig. 2 shows the sequence of process "NXDOMAIN" responses, and Fig. 3 shows the sequence of process 'NXERROR' responses.

[2] https://www.thc.org/thc-ipv6/

We have modified the program to gather data for our study. We used a Poisson distribution with a mean time between queries of 1 s so as to minimize the load on DNS servers. Also, we embed information about our survey into every reverse lookup request, in order to explain our survey to network administrators.

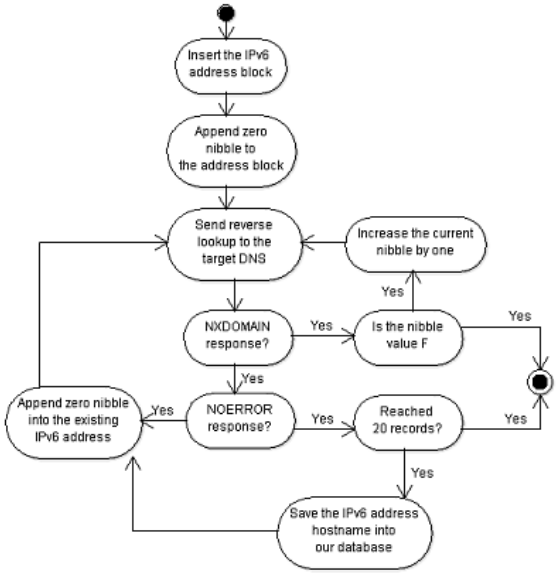

Fig. 1. Activity diagram of reverse search algorithm.

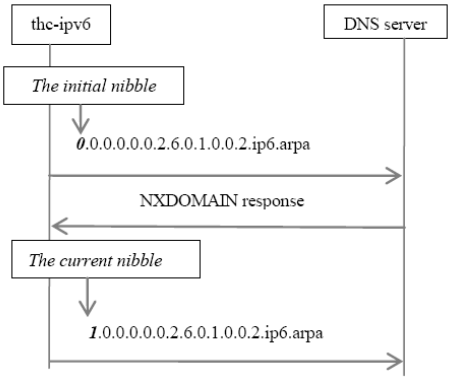

Fig. 2. The sequence of process 'NODOMAIN' responses.

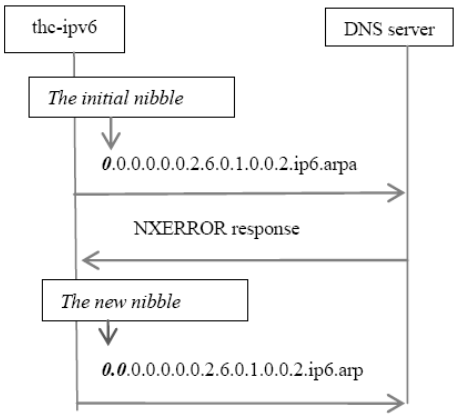

Fig. 3. The sequence of process 'NXERROR' responses.

IV. Observations

We set up our survey in January 2014 and collected the results from fifty countries. The survey results help us to better understand the potential problems of some existing IPv6 address allocation mechanisms. RFC 3177 [13] recommends RIRs assign /48 address blocks to each registered organization. Therefore, to have a complete IPv6 address, network administrators need only allocate the Subnet and Interface ID fields. From our survey results, we have observed that some countries have a larger number of allocated address prefixes, but only few of them have been assigned to end users. Around 85% of sites have more than one subnet, i.e. network administrators place different values in the Subnet ID field. However, 15% of sites show that all hosts are located in the same network, and network administrators have assigned zero values into the Subnet ID field. Moreover, we observe that 75% of IPv6 addresses have long runs of zero bits.

TABLE I: The Arrangement of Channels

IPv6 Address	Host Name
2001:200:0:2::800:1	lo-0.hitach2.nara.wide.ad.jp
2001:200:0:2::800:3	lo-0.juniper4.nara.wide.ad.jp
2001:200:0:2::1800:1	hitachi1.otemachi.wide.ad.jp
2001:200:0:2::1800:5	lo-1.foundry6.otenachi.wide.ad.jp

A. Common Patterns in Interface IDs

By analyzing the Interface ID field from our survey results, we noticed five existing IPv6 address allocation mechanisms that could give rise to security issues. The following sections consider how those mechanisms are used to generate IPv6 addresses.

1) Sequential increase host numbering: The majority of results show that the values in the Interface ID field have been sequentially increased when generating a new IPv6 address.
2) Use bit fields as subfield identifiers: Some network administrators allocate four non-zero bytes to the Interface ID field, so as to classify different groups in the same subnet. In our results, we observe that two bytes from the Interface ID field indicate the group ID, and the other two bytes identify hosts in this group. For example, Table I shows how to divide hosts from the same subnet into different groups.
3) Use IPv6's stateless auto-configuration mechanism: Basically, the Interface ID field is derived from its 48-bit MAC address by inserting FF: FE into its middle. Since the first three bytes of the MAC address represent the interface's Organizationally Unique Identifier (OUI), if the network interface cards were made by the same company, their Interface IDs have the same OUI.
4) Use a domain's IPv4 address to fill the Interface ID field: Each field of an IPv6 address contains 16 bit values and is represented by four hexadecimal digits. However, we observed that network administrators from Norway decided on a different way to allocate the Interface ID field; they have used four 16- bit parts of the Interface ID field to represent the 8-bit integers of an IPv4 address.
5) Transition allocation mechanisms: During the IPv4 and IPv6 coexistence period, some transition methods have used special mechanisms to allocate the IPv6 address, such as 6to4, IVI and Teredo. One of our goals for this research was to observe such transition allocation mechanisms, but we haven't found this type of mechanism in our survey results.

B. Survey Results Summary

The aim of this survey is to better understand how the IPv6 addresses are allocated in actual practice. Selecting the survey data is difficult. There are three principles for our data selection: the selected countries have a large number of assigned IPv6 address prefix, each country has IPv6 domain records in its DNS servers and each domain includes some IPv6 addresses.

TABLE II: COLUMNS SHOW NUMBER OF IPV6 DOMAINS OBSERVED IN EACH COUNTRY

Country	Assigned Address prefixes	Total DNS Domain	DNS Domain has more than 10 addresses records	Sequential Integers		Subfield Identifies		Auto Configuration		IPv4 addresses		No obvious pattern	
				10..20	>20	10..20	>20	10..20	>20	10..20	>20	10..20	>20
United States	832	74	38	10	24	0	0	1	2	0	0	0	1
Sweden	197	47	20	5	12	0	0	0	2	0	0	0	1
Germany	180	34	19	8	10		0	0		0	1	0	
United Kingdom	107	28	25	3	17	1	0	0	2	0	0	0	2
Russian Federation	108	23	13	3	7	0	0	0	1	1	0	1	0
Australia	194	21	10	3	6	0	0	1	0	0	0	0	0
Netherlands	124	21	19	4	14	0	0	0	0	0	1	0	0
Czech Republic	73	19	8	2	5	0	0	0	0	0	0	1	0
France	64	19	13	2	9	0	0	1	0	0	0	1	0
Ukraine	65	16	14	4	7	0	0	0	0	1	0	1	1
Poland	94	15	9	1	6	0	0	0	0	0	0	1	1
Switzerland	69	15	13	4	5	0	0	0	0	2	0	2	0
Austria	85	14	4	2	1	0	0	0	0	0	0	1	0
Brazil	185	14	13	3	8	0	0	2	0	0	0	0	0
Argentina	52	14	8	2	3	0	0	2	0	0	0	1	0
Indonesia	122	14	8	2	4	0	0	1	0	0	0	1	0
Canada	115	12	9	1	7	0	0	0	0	0	0	1	0
Belgium	26	11	5	2	1	0	0	1	0	0	0	1	0
Norway	40	11	8	1	4	1	0	1	0	0	0	1	0
Slovenia	47	10	7	4	2	0	0	0	0	0	0	1	0
Total	2779	432	263	218		2		17		6		20	

From the RIR statics site [10], we reviewed 214 countries around the world. We found the countries with significant IPv6 activity by looking at their number of assigned: /48 address blocks. Theoretically, if the country has a large number of assigned IPv6 address prefixes, it also has more DNS domains to map those prefixes. However, the results from the top fifty countries indicate that some countries have many assigned address prefixes with few IPv6 domains. For example, India has 80 assigned IPv6 address prefixes, but no observed DNS domains. We assume that some countries are still deploying IPv6, so they don't yet have the IPv6 domains information in their DNS servers. Our survey methodology uses reverse DNS requests to find IPv6 domains for each address prefix; therefore, we select the countries that have more DNS domains. We have chosen the top twenty countries that have more than ten DNS domains. Those domains address prefixes will be used as the input parameters for the program.

Table II displays the observed usage of IPv6 allocation mechanisms in each country. It indicates that some domains have few IPv6 records in the DNS server; we surmise that those results represent a test or small sites providing IPv6 services. Other network administrators may have only a few IPv6 addresses for users to access, it is not a compulsory requirement to set up a reverse DNS domain for every IPv6 addresses in the network.

For each IPv6 allocation method, we list two columns: "10..20" represents domains with 10 to 20 IPv6 addresses; ">20" indicates domains with more than 20 IPv6 addresses. Moreover, some domains have less than ten IPv6 addresses, we do not include such domains in our table.

All in all, in order to understand the IPv6 allocation mechanisms usage, we use the number of observed domains for each method to divide the domains that have more than 10 IPv6 address records.

Table II shows that for the 263 domains observed with 10 or more IPv6 addresses, 82% domains use sequential numbers in the interface ID field for their IPv6 addresses. 7% of domain's IPv6 addresses are allocated by using the stateless auto-configuration mechanism, and only 2.2% domains mostly European countries show that network administrators have filled their existing IPv4 addresses into the Interface ID field. Less than 1% of domains use bit fields as subfield identifiers, indicating that hosts in the same subnet have been subdivided into different groups. We also notice that 7.6% of domains don't use these special patterns to fill the Interface ID field for IPv6 addresses. These results imply that some network administrators show concern for their network security and user privacy. In the next section, we summarize our observations and briefly discuss our future

work.

V. Conclusion and Future Work

By finding the most common existing IPv6 Interface ID field allocation mechanism, and surveying their usage in different countries, we found that

- It is feasible to launch an effective network scanning attack in the existing IPv6 network, because many network administrators are not allocating their Interface ID fields with non-predictable values. Our survey results show that if the hosts belong to the same group, then network administrators prefer to allocate IPv6 addresses with some common patterns or use sequential numbers in the Interface ID field. We remark that network administrators prefer to use meaningful values in the Interface ID field; we assume this will help them identify a host machine when something goes wrong, for example, some network administrators use an existing IPv4 address in their Interface ID fields.
- In [1], Chown mentioned that it is better not to use MAC addresses in EUI-64 format. Doing so helps an attacker to reduce searching time if the attacker knows the hardware brand for many hosts in a site. However, we observed that 7% survey results have used MAC addresses and FF: FE patterns to fill the Interface ID field.
- It is not a compulsory requirement to set up a reverse DNS domain in the IPv6 network. From our survey, we observe that some hosts have IPv6 addresses, but no AAAA records in their domain's DNS server. In such cases, our methods will not gather IPv6 addresses from the DNS server.

RFC 5157 [1] has summarized some ways for preventing IPv6 network scanning attacks. However, in the existing IPv6 network, Network Intrusion Detection System (NIDS) tools are unlikely to detect the reverse DNS lookup search mechanism discussed in this paper. In the future, we plan to develop rules for some existing NIDS tools, such as Bro and Snort, which will detect this type of address scanning. Once scanning attacks have been observed, the NIDS will drop or log the information based on the rule setting.

Acknowledgment

This survey is based upon work supported by the University of Auckland and Tsinghua University IPv6 research group; our colleagues from Tsinghua University helped with setting up the survey environment and answered questions on survey results from China.

References

[1] T. Chown, "IPv6 implications for network scanning," Internet RFC 5157, 2008.
[2] T. Aura, "Cryptographically generated addresses (CGA)," Internet RFC 3972, 2005.
[3] A. Alsadeh, H. Rafiee, and C. Meinel, "Cryptographically Generated Addresses (CGAs): Possible attacks and proposed mitigation approaches," in *Proc. 12th IEEE International Conference, Computer and Information Technology*, Chengdu 2012, pp. 332-339.
[4] R. Hinden, M. O'Dell, and S. Deering, "An IPv6 aggregatable global unicast address format," Internet RFC 2374, 1998.
[5] A. Cooper, S. Farrell, and S. Turner, "Privacy requirements for IETF protocols, draft-cooper-ietf-privacy-requirements-00," 2013.
[6] M. Crawford, "Transmission of IPv6 packets over ethernet networks," internet RFC 2464, 1998.
[7] B. Carpenter, G Van de Velde, T. Hain, R. Droms, and E. Klein, "Local network protection for IPv6," Internet RFC 4864, 2007.
[8] E. Davies, S. Krishnan, and P. Savola, "IPv6 transition/coexistence security consideration," Internet RFC 4942, 2007.
[9] D. Barr, "Common DNS operational and configuration errors," Internet RFC 1912, 1996.
[10] Regional Internet Registries Statistics. [Online]. Available: http://www-public.int-evry.fr/~maigron/RIR_Stats/RIR_Delegations/World/IPv6-Alpha.html
[11] P. Mockapetris, "Domain names - concepts and facilities," Internet RFC 1034, 1987.
[12] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi, "DNS extensions to support IP Version6," Internet RFC 3596, 2003.
[13] IAB. IAB/IESG Recommendation on IPv6 Address Allocations to Sites. Internet RFC 3177, 2001.

Qinwen Hu received the B.A degree in computer science and information system from the Massey University, Auckland, New Zealand. And the master degree in computer science from the Auckland University, Auckland, New Zealand.

He is currently a Ph.D. candidate at computer science department. The University of Auckland, Private Bag 92019, Auckland 1142, New Zealand. His research interests include IPv6 QoS, network security and traffic measurement in the IPv6 network.

Nevil Brownlee has been an associate professor in the University of Auckland's Computer Science Department since 2004; his teaching and research focuses on the Internet, especially on Internet data collection and Measurement and more recently the analysis of unsolicited traffic. Nevil managed the University's campus network from its beginnings in 1985, its connection to the Internet in 1989, and its further development to about 1998, thus gaining experience in operating a medium-sized (14,000 hosts) network.

Since 2000 Nevil has been associated with CAIDA (the cooperative association for internet data analysis); his work there includes measurement and analysis of Internet traffic flows, and of the behaviour of the global Domain Name System (DNS). He is also associated with the WAND network research group at the University of Waikato. Nevil has taken an active part in the IETF since 1992. From 2000 to the present he has been co-chair of the IPFIX (IP Flow Information Export) Working Group, and from 2012 he has co-chaired the EMAN (Energy Management) Working Group. As well, since February 2010 he has been the RFC editor for independent submissions; he is also a member of RSOC, the IETF's RFC series oversight Committee.