On the Generation of Fast Verifiable IPv6 Addresses

Qianli Zhang and Xing Li

Tsinghua University, Beijing 100084, China
zhangql02@mails.tsinghua.edu.cn

Abstract. Many network attacks forge the source address in their IP packets to block traceback. This situation does not change much in IPv6 network since IPSEC is not enabled generally and most IP address spoof attacks have taken effect before packets reached destination. Although ingress filtering can be used to validate source addresses, it could only ensure that the network portion of an address is not spoofed. Since subnets are much larger in IPv6, even with RFC 2827-like filtering an adversary can spoof an enormous range of addresses. In this paper, we propose an IPv6 address assignment scheme to generate verifiable IPv6 addresses in one network. With this scheme, router could validate the IPv6 addresses quickly, thus allow all outgoing packets with improper source addresses and all incoming packets with improper destination addresses to be immediately identified. Apart from the obvious merit to counter denial of service attacks, this scheme also make network audit and pricing easier.

1 Introduction

Attackers commonly forge source addresses to hinder tracing of their malicious packets. Examples include DDoS attacks [1], smurf attacks[2], and TCP SYN flooding attacks[3]. Reliably detecting the attacker is hard because standard routers cannot verify that a packet is indeed sent by the node specified in its source address. Ingress filtering[4] is widely used to validate source addresses. RFC 2827 specifies methods to implement ingress filtering to prevent spoofed traffic at its origin. Unfortunately such filtering lacks of the initiative for the origin network to implement. Also RFC 2827 ensures that only the network portion of an address is not spoofed, not the host portion. For example, for 24-bit subnet 192.0.2.0/24, RFC 2827 filtering ensures that traffic originating from 192.0.3.0 is dropped but does not stop an adversary from spoofing all the hosts within the 192.0.2.0/24. Since subnets are much larger in IPv6, even with RFC 2827-like filtering an adversary can spoof an enormous range of addresses. Currently no techniques are available to mitigate the spoofing of the 64 bits of host address space available in IPv6.

Another approach to the problem of IP spoofing is tracing[5]. Since source addresses are unreliable, tracing requires expensive and complicated techniques to observe traffic as they pass through routers and reconstruct a packets travel

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part IV, LNCS 3994, pp. 176–179, 2006.

path at the end. Tracing also becomes ineffective when the volume of attack traffic is small or the attack is distributed. Moreover, tracing is typically performed after an attack is detected, and perhaps the victim has already been damaged.

In this paper, we propose a scheme to assign verifiable IPv6 addresses in a network. With this scheme, router could validate not only the subnet part but also the interface part of the IPv6 addresses quickly. Apart from the obvious value in ingress filtering, this scheme can also ensure that incoming packets with improper destination addresses to be immediately identified and dropped. Thus it provide some initiative for its deployment. With identifier contained in the addresses, it could also be used to identify the possible sources of an attack. Intrusion detection and network problem diagnosis can also be simplified.

This paper is structured as follows: Section 2 presents some background notations and information. Details are provided in section 3. The paper concludes in section 4.

2 Background Notations

An IPv6 address is 128 bits long. It is divided into two parts. The leftmost 64 bits, the subnet prefix, is used for routing IP packets across the Internet to the destination network. The rightmost 64 bits, the interface identifier, identifies an individual node within a local network. The interface identifiers may be chosen in an arbitrary way, e.g. randomly, as long as no two nodes on the same network share the same value.

Two bits of the interface identifier have a special semantics. The 7th bit from the left is the Universal/Local bit or ”u” bit. It is usually set to 1 to mean that the interface identifier is configured from an EUI-64 identifier from the interface hardware and, thus, is globally unique. The 8th bit from the left is the Individual/Group or ”g” bit, which is set to 1 for multicast addresses.

To better present our scheme, the following notations are used throughout the paper.

- hash: Cryptographic hash function, SHA-1[6] for example.
- hashT: Cryptographic hash function whose output is truncated by taking the T leftmost bits of the output.
- cipher64: 64 bits block cipher, IDEA[7] for example.

3 Verifiable IPv6 Address Generation

The verifiable address is generated by a local authority, DHCP server for example. In figure 1, ID is the identifier generated by authority for tracking, R is not used now and will be set to zero. P is prefix required and used to generate destination specific addresses. Given the destination address D, prefix requirement P_r, subnet prefix N and the correspondent key K_N, the generation procedure is as follows.

1. set $Padding$ and R to 0, P to P_r, ID to the identifier.
2. $SIG = HASH32((D >> (128 - (P_r >> 3)))|N|R|P_r|ID)$, where | is concatenation.

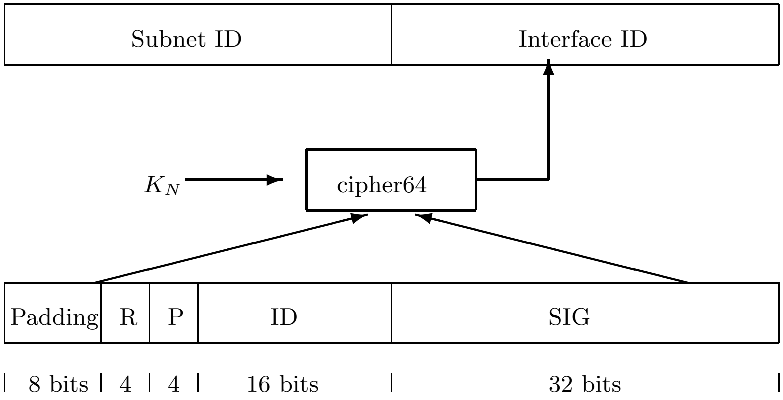

Fig. 1. Verifiable IPv6 address

3. encrypt the 64 bits by *cipher*64 with key K_N. Encryption guaranteed the generated addresses can not be discriminated from randomly generated addresses easily. Encryption also provide a method to keep the ID information confidential.
4. test whether "u" bit is 0 and "g" bit is zero. If not, increase padding by 1 and repeat the last step. Since a total of 256 IPv6 interface ID could be generated, the probability of this 256 IPv6 addresses all have the "u" bit to 1 or "g" bit to 1 is $(\frac{3}{4})^{256}$, which is about 10^{-32} and negligible.

Required prefix mandates when the calculation of SIG also includes the leftmost P_r bytes of destination IPv6 address. When P_r is not zero, the address generated is destination specific and could not communicate to hosts out of the range. If P_r is zero, the generated address is a static IPv6 addresses and can communicate to all IPv6 addresses. Destination specific addresses make IP address spoof even harder since even if an attacker knows a valid IPv6 address, he could not decide whether this address is a static one. The limitation of destination specific addresses is, however, that it requires to extend DHCPv6 protocol. Also, the value of destination specific addresses is limited since servers have to have static addresses.

The verification procedure is similar.

1. decrypt the 64 bits interface ID by cipher64 with key K_N. if R is not zero, discard the packet.
2. set *Padding* to 0, $SIG_v = HASH32((D >> (128 - (P_r >> 3))) |N|R|P_r|ID)$, where | is concatenation, if SIG_v does not equal SIG, discard the packet.

For a large domain, it may be of interest to generate K_N with a master key K_m. For example, $K_N = hash(K_m|N)$. However, this scheme does not mandate the specific method to generate K_N.

In this scheme, only symmetric cryptography is used, which make it scalable for the high speed filtering. Asymmetric cryptographic primitives, such as RSA signatures[8], are computationally expensive: RSA signature verification is about three orders of magnitude slower than one symmetric operation (block cipher or hash function operation), and signature generation is about four orders of magnitude slower. When implemented in hardware, the speed difference is even larger. Thus make this algorithm feasible for high-speed implementation.

4 Conclusion

In this paper, a new scheme to generate verifiable IPv6 addresses is introduced. This scheme make the IPv6 host portion ingress filter feasible. Also since only symmetric cryptography is used, this scheme could be implemented in routers and provide better protection for network bandwidth DOS.

More research is required to resolve the following problems. First of all, for large organizations, it is often desirable to have a key management protocol to deal with the key generation and distribution. Secondly, the process of destination specific addresses generation is worth further research.

References

1. Computer Emergency Response Team. CERT Advisory CA-2000-01 Denial-of-Service Developments, http://www.cert.org/advisories/CA-2000-01.html, January 2000.
2. Computer Emergency Response Team. CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks, http://www.cert.org/advisories/CA-1998-01.html, January 2000.
3. C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on TCP, Proceedings of IEEE Symposium on Security and Privacy, 1997.
4. P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, RFC 2827, May 2000.
5. Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, Network Support for IP Traceback, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 9, NO. 3, JUNE 2001
6. C. Madson and R. Glenn, The Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November 1998.
7. A. J. Menezes, P.C. v. Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press New York, 1997, p. 265.
8. R. L. Rivest, A. Shamir, and L. M. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2):120-126, 1978.