Network Working Group H. Rafiee
INTERNET-DRAFT Ciber AG
Intended Status: Informational Track C. Meinel
Hasso Plattner Institute
Expires: August 8, 2014 Februar 8, 2014

Possible Attack on Cryptographically Generated Addresses (CGA)
<draft-rafiee-6man-cga-attack-01.txt>

Abstract

This document describes the new vulnerability with the use of Cryptographically Generated Addresses.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 4, 2014.

Table of Contents

1. Introduction

Cryptographically Generated Addresses (CGA) [RFC3972] is one of the important options of Secure Neighbor Discovery (SeND) [RFC3971] in IPv6 networks. CGA provides the node with the proof of IP address ownership by finding a binding between the public key and the node's IP address. Therefore, It can protect the nodes from network layer IP spoofing attack and prevent forging the identity (if it is only based on the IP address). However, CGA, itself is vulnerable to some types of attacks such as DoS, replay attack (The use of timestamp would mitigate this attack), etc [3]. The goal of this document is not to focus on the well-known attacks but the new CGA vulnerabilities.

2. Sec value vulnerability

CGA values are the fingerprint of public key. They are generated by executing a hash function on public key and some other parameters. Since the default algorithm for generating this hash is SHA-1, the attacker node only needs to do brute force attacks against 59 bits. CGA algorithm uses sec value (a value between 0 to 7) to increase the brute force search space from 59 bits to maximum 171 bits (59+sec*16) and as a result complicates the brute force attacks to break CGA. Nevertheless, in practice, only sec value 0 and 1 can be used because it takes hours to years to generate CGA sec value higher than 1 [2].

Unfortunately, in practice, it does not matter what sec value the victim node chooses and the use of sec value only complicates the IP address generation process for the victim node. This is because the attacker will only use sec value 0 and SHA1 algorithm.

The reason are as follow:

- No comparison of source address and target address

Based on the Neighbor Discovery Protocol (NDP) specification on section 7 RFC 4861 [RFC4861, RFC4862], there is nothing about to compare the source IP address with the target address. In SeND specification [RFC3971], there are rules for the sender node. However, the verifier node never checks those rules. This is why the attacker can ignore them. So, the attacker can create the SeND message by using his own CGA address that differs only in sec value. The attacker selects the victim node's source address as his own target address and sends this message.

- The CGA verifier node ignores 3 bits sec value in source address and 2 bits u and g

Based on NDP specification, the verifier node checks to see whether or not the target address is the same as its own IP address. If it is the same and the node supports CGA, then it starts CGA verification. Based on step 4 section 5 RFC 3972, the CGA node compares the source address (IID section) of the sender node to his own IID. The verifier node ignores 3 bits sec value. So, the attacker can set the target address to the real CGA address of the victim node disregard its sec value and set the source address to his own CGA value that is only different in the 3 leftmost bits. Since the verification is successful, the attacker can spoof the IP address of CGA node.

- Either conflict on the network or the CGA node waive his rights on the IP address

The attacker node can persist on his own IP address after a successful verification by CGA node and either force CGA node to generate a new IP address and again the attacker repeats this process or there will be duplicate addresses on the network which cause many services in the victim network stop working. This is because all the nodes verify this attacker node the same way as the legitimate CGA node processed the verification. From their aspects, these two nodes are the same.

The mentioned flaw occurs during verification processes in all verifier nodes. The node needs to verify other nodes in two different conditions -- during DAD process and during checking the neighbors' reachability in cache. This means that the CGA security is only the security of CGA sec value 0 that is 2^59 bits.

- The lower limit for key size is 384 bits

The attacker does not need to worry about attack on public key and he can choose the lowest size public key so that he can better play with the RSA values and easier and faster generates the similar hash of the CGA node.

- Modifier can be zero

The attacker does not need to generate a really good random value. Since for him it is only important to match the hash value. This is especially true for the scenario where the attacker needs to do brute force attacks against all 64 bits and sec value is not ignored.

In the following subsections, some of these attacks are explained in more detail.

2.1. Duplicate Address Detection Process

When a node generates his IP address, it process the DAD in order to

avoid collision on the network. The attacker might be able to generate the CGA value the same of the legitimate CGA node and claim the ownership of that IP address. The CGA nodes only tries 3 times and then it gives up.

2.2. Nodes communications

When two nodes want to start communication, they try to find the IP address of eachother by sending multicast NS/NA messages. If the attacker can generate the CGA of one of these nodes, he can spoof the identity of them. This is what against the CGA goal.

3. Security Considerations

-

4. IANA Considerations

-

5. Appendix

- CGA multicore attack

This is where you can find CGA attacks (multicore). More attacks will uploaded in the following link:

http://www.hpi.uni-potsdam.de/meinel/security_tech/ipv6_security/ipv6ssl.html

6. Acknowledgements

The author would like to acknowledge Fabian Braeunlein, one of a bachelor student at Hasso Plattner Institute who assists us, during this busy moments, for writing the attacking codes.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to
Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)," RFC 3972, March 2005.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., Soliman, H., "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] Thomson, S., Narten, T., Jinmei, T., "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[1] AlSa'deh, A., Rafiee, H., Meinel, C., "Cryptographically Generated Addresses (CGAs): Possible Attacks and Proposed Mitigation Approaches," in proceedings of 12th IEEE International Conference on Computer and Information Technology (IEEE CIT'12), pp.332-339, 2012.

[2] Bos, J., Oezen, O., Hubaux, J., "Analysis and Optimization of Cryptographically Generated Addresses", In Proceedings of the 12th International Conference on Information Security (2009), ACM, pp. 17 ? 32.

[ugbits] Carpenter, B., Jiang, S., "Significance of IPv6 Interface Identifiers", http://tools.ietf.org/html/draft-ietf-6man-ug, November 2013

Authors' Addresses

Hosnieh Rafiee
Ciber AG
KoelnTurm
Im Mediapark 8
http://www.ciber.com
Phone: +49 (0221) 272 67- 122
Email: ietf@rozanak.com

Christoph Meinel
Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
Email: meinel@hpi.uni-potsdam.de