Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 7, July 2016, pg.529 – 534

REMOVING IPV6 SECURITY USING HYBRID ALGORITHM

Mamta Bhatia, Archana
M.Tech (CSE), Assistant Professor (CSE)
PM College Of Engineering, Kami Road, Sonepat
rathee.sandhya@gmail.com, asandhu43@gmail.com

Abstract*— The default method for IPv6 address generation use an organizationally Unique Identifier(OUI) assigned by the IEEE Standards Association and an Extension Identifier assigned by the hardware manufacturer. For this reason a node will always have the same Interface ID whenever it connects to a new network. Because the node's IP address does not change, the node will be vulnerable to privacy related attacks. Currently this problem is addressed by the use of two mechanism that do not use MAC addresses or other unique values for randomizing the IID during its generation: Cryptographically Generated Addresses (CGA) and privacy Extension. The problem with the former approach is the computational cost involved in the IID generation and, more importantly, the verification process. The problem with the later approach is the lack of necessary security mechanisms and that it provides the node with only partial protection against privacy related attacks. To enhance the security in the IPv6 address we use well known cryptographic algorithm called Diffie hellman and blowfish.

I. INTRODUCTION

IPV6 was firstly introduced by IETF (Internet Engineering Task Force) in mid 1990's. IPV6 is a next generation protocol that tries to overcome the problems due to IPV4. IPV6 provides 128-bit address space that is 3.4*(10)^38 addresses. This address space is very large(its in trillions in trillions). As we all are aware of the use of internet enable resources worldwide so the need of IP addresses are increasing day by day. That results in the deployment of IPV6. Because the addresses provided by IPV4 are only 4,294,967,296 (4 billion) and have been used almost. Several experts forecast that IPV4 will be finished completely in upcoming years because of insufficient addressing space so the migration from IPV4 to IPV6 is necessary to meet the requirement of future network. As we are trying to migrate from IPV4 to IPV6,there are some security issues that arise. Some are due to IPV4 and some are due to IPV6. Firstly we will define the features of IPV6, secondly identify the vulnerabilities due to IPV6 and then use some technologies to remove those vulnerabilities.

II. IPV6 SECURITY ISSUES

There are lots of security risks and threats occur in the deployment of IPV6 protocol.

1. Reconnaissance attack - Attackers may get information about host and network devices and their interconnection in the targeted network by using two methods- ACTIVE and PASSIVE methods. In the active method intruders do scanning of the data and in the passive method they fetch the essential data about the enterprise network.
2. Extension Header - Long chain of headers make a security device difficult to do deep packet inspection in the transport layer header and will increase as the malicious node will fragment the packet into very small size. Thus it will force the security device to reassemble those small packets before inspection.
3. Denial Of Service(DoS) Attack - As intruders split the packets into small size of fragments so it will send large number of fragments to the target system until it become overload and crash the system.
4. Malicious router - As IPV6 use SLAAC for autoconfiguration of IP address so a malicious router may decide to serve as a legitimate router and misguides the packets in the network.

III. VULNERABILITIES

Vulnerabilities of IPv4 Because of its end-to-end model, IPv4 hasn't any security implemented. It completely relies on the hosts to provide security. As a result of this implementation it has a numerous amount of security threats which has become well known over the years. The most common and well known threats are:

• *Viruses, Trojans and Worms:* These types of malicious programs can spread themselves from one infected hosts to another. Although the words Virus, Trojan and worm are used interchangeably, they are not the same thing. A virus attaches itself to a file enabling it to spread from one computer to another, leaving infections as it travels. In a way the worm is similar to a virus and also spread from computer to computer but it has the capability to spread without human action. A Trojan will appear to be useful software but will do damage once installed or run on the computer. It is mostly known to make a backdoor to the infected computer. Trojans are not automatically spread from computer to computer.

• *Port scanning and reconnaissance:* This is the process of scanning a host to determine which TCP and UDP ports are accessible. Open ports can be used to exploit the specific hosts further.

• *Fragmentation attacks*: The basic modus operandi of IP fragmentation attacks is to use varied IP datagram fragmentation to disguise its TCP packets from a target's IP filtering devices. For example the ''ping of death'' attacks. This attack uses many small fragmented ICMP packets which when reassembled at the destination exceed the maximum allowable size for an IP datagram which can cause the victim host to crash, hang or even reboot.

• *Man-in-the-middle attacks (MITM)* : It is a form of active eavesdropping where the the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

• *Denial of Service Attacks (DoS)*: DoS attacks are implemented by either forcing the targeted computer to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately. Many techniques or method had been developed to overcome the abovementioned security issues. For instance, the use of 'IPSec' to aid the use of encrypted communication between hosts, but this is still optional and continues to be the main responsibility of the end hosts.

IV. Blowfish

Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all uses. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. Blowfish is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches.

Blowfish is a symmetric block encryption algorithm designed in consideration with,

- **Fast:** It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte.
- **Compact:** It can run in less than 5K of memory.
- **Simple:** It uses addition, XOR, lookup table with 32-bit operands.
- **Secure:** The key length is variable ,it can be in the range of 32~448 bits: default 128 bits key length.

1. Description of Blowfish Algorithm:

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time.it will follows the feistel network and this algorithm is divided into two parts.

1. Key-expansion
2. Data Encryption

3. Key-expansion:

It will converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Blowfish uses large number of subkeys.

These keys are generate earlier to any data encryption or decryption.

The p-array consists of 18, 32-bit subkeys:

P1,P2,……,P18

Four 32-bit S-Boxes consists of 256 entries each:

S1,0, S1,1,……. S1,255

S2,0, S2,1,…….. S2,255

S3,0, S3,1,…….. S3,255

S4,0, S4,1,.....S4,255

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

4. **Data Encryption:**

It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round. Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a keydependent permutation, and a key- and data-dependent substitution.

2. Algorithm:Blowfish Encryption:

Divide x into two 32-bit halves: xL, xR

For i = 1to 16:

xL = XL XOR Pi

xR = F(XL) XOR xR

Swap XL and xR

Swap XL and xR (Undo the last swap.)

xR = xR XOR P17

xL = xL XOR P18

Recombine xL and Xr

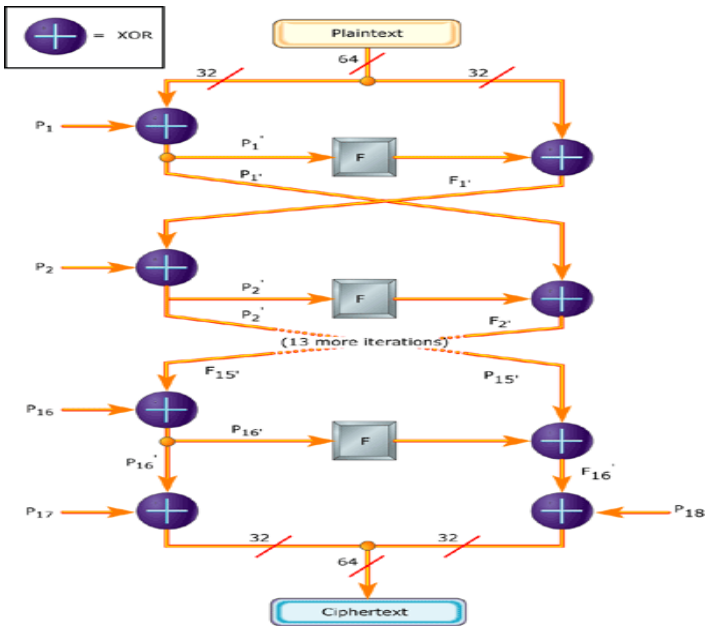

Figure 1.7.2: Blowfish Encryption

5. Diffie Hellman

In [DH76] Diffie and Hellman describe a method for two parties to agree upon a shared secret in such a way that the secret will be unavailable to eavesdroppers. This secret may then be converted into cryptographic keying material for other (symmetric) algorithms. Diffie-Hellman key agreement requires that both the sender and receiver of a message have key pairs. By combining one's private key and the other party's public key, both parties can compute the same shared secret number.This number can then be converted into cryptographic keying material. That keying material is typically used as a key-encryption key (KEK) to encrypt a content encryption key (CEK) which is in turn used to encrypt the message data

6. Diffie–Hellman key exchange (D–H)

D-H is a cryptographic that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Synonyms of Diffie–Hellman key exchange include:

- Diffie–Hellman key agreement
- Diffie–Hellman key establishment
- Diffie–Hellman key negotiation
- Exponential key exchange
- Diffie–Hellman protocol

The Diffie-Hellman key agreement was invented in 1976 during a collaboration between Whitfield Diffie and Martin Hellman and was the first practical method for establishing a shared secret over an unprotected communications channel. Ralph Merkle's work on public key distribution was an influence.

The method was followed shortly afterwards by RSA another implementation of public key cryptography using asymmetric algorithms.

In 2002, Martin Hellman wrote:

> The system...has since become known as Diffie-Hellman key exchange. While that system was first described in a paper by Diffie and me, it is a public key distribution system, a concept developed by Merkle, and hence should be called 'Diffie-Hellman-Merkle key exchange' if names are to be associated with it. I hope this small pulpit might help in that endeavor to recognize Merkle's equal contribution to the invention of public key cryptography.
>
> US Patent 4,200,770, now expired, describes the algorithm and credits Hellman, Diffie, and Merkle as inventors.

V. CONCLUSION

The problem with the latter approach is the lack of necessary security mechanisms and that it provides the node with only partial protection against privacy related attacks. To enhance the security in the IPV6 address we use our proposed system. As it is known that privacy is an important issue in present time because of number of attacks in the network. So the best method for

securing a network is generating random interface identifier so that intruders can not track the IP address easily and data can be secured. Methods for generating random ID are CGA, Privacy Extension Method and SSAS. Here some techniques in which EUI-64,CGA and Privacy Extension has limitation and some are good enough like SSAS and i-SeRP. SSAS takes less time to remove the vulnerabilities in comparison to CGA. But still SSAS has limitation because it takes a long computational time for processing. while i-SeRP calculate the risk value and then decide to use right model to counter the risks. In the proposed solution as 128 bit unique address is generated so it will prevent the malicious nodes to enter in the network and make the network secure. Because in the proposed work 'certification authentication' is used for preventing malicious node. And secrets will be exchanged by 'Diffie hellman Key Exchange Algorithm', and the key will be encrypted by BlowFish Encryption Algorithm. And to know the existence of malicious nodes, periodically challenges will be send. So it is more secure than other described method.

REFERENCES

[01] Claude Castelluccia, Gabriel Montenegro, Julien Laganier and Christoph Neumann," Hindering Eavesdropping via IPv6 Opportunistic Encryption", P. Samarati et al. (Eds.): ESORICS 2004, LNCS 3193, pp. 309–321, 2004.

[02] Eng. N Pradeep Ruwan Nawarathne," Overhead of FTPS and FTP over IPsec in IPv6 networks", International Journal of Scientific & Engineering Research, Volume 3, Issue 11, November-2012

[03] Steffen Hermann and Benjamin Fabian," A Comparison of Internet Protocol (IPv6) Security Guidelines", Future Internet 2014,

[04] Priya Tayal," IPV6 SLAAC related security issues and removal of those security issues", International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 3 Issue 9, September 2014 Page No. 8445-8459

[05] Lokesh Galla , Suyesh Regmi," IPv4-IPv6 Transition Techniques", Technical report, May 2011

[06] *HyunGon Kim and Jong-Hyouk Lee," Diffie-Hellman Key Based Authentication in Proxy Mobile IPv6",Mobile Information Systems Volume 6 (2010),*

[07] *Martin Ehmke and Kaj Grahn and Jonny Karlsson," Securing Control Signaling in Mobile IPv6 with Identity-Based Encryption", Issues in Informing Science and Information Technology Volume 6, 2009*

[08] *Tina Sharma and Savita Shiwani," Statistical Results of IPSec in IPv6 Networks", International Journal of Computer Applications (0975 – 8887) Volume 79 – No.2, October 2013.*

[09] *Hero Modares a,n , Amirhossein Moravejosharieh b , Jaime Lloret c , Rosli Salleh," A survey of secure protocols in Mobile IPv6", Journal of Network and Computer Applications & 2013*

[10] *Muhammad Zubair, Xiangwei Kong, Saeed Mahfooz, and Irum Jamshed," SIDP: A Secure Inter-Domain Distributed PMIPv6", International Journal of Information and Electronics Engineering, Vol. 4, No. 2, March 2014*

[11] *Thorsten Aurisch Christoph Karg," Using IPSec for Secure Multicast Communications", International Command and Control Research and Technology Symposium 2003*

[12] *Qiu Ying and Bao Feng," Authenticated Binding Update in Mobile IPv6 Networks", 2010 IEEE*

[13] *Fabien Allard and Jean-Marie Bonnin," IKE Context Transfer in an IPv6 Mobility Environment", MobiArch'08, August 22, 2008, Seattle, Washington, USA*

[14] *M. Anand Kumar and Dr. S. Karthikeyan," Security Model for TCP/IP Protocol Suite", JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY, VOL. 2, NO. 2, MAY 2011*

[15] *Hengky Susanto and Byung Guk Kim," Functional Scheme for IPv6 Mobile Handoff", International Journal of Computing and Network Technology 2014*

