Initialization Vector Attacks on the IPsec Protocol Suite

Christopher B. McCubbin Ali Aydın Selçuk Deepinder Sidhu

Maryland Center For Telecommunications Research
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250, USA
E-mail: {cmccub1,aselcu1,sidhu}@umbc.edu

Abstract

In this paper, we analyze the security of IPsec against a class of attacks known as the IV attacks, which are based on modifying the initialization vector (IV) of a CBC-encrypted packet during transmission. We show that IV attacks can be a serious threat for IPsec if IPsec is not used carefully. We also discuss the defense methods against these attacks.
Keywords. *Internet security, IPsec, IV attacks, cryptanalysis.*

1. Introduction

IPsec is a security protocol suite that provides encryption and authentication services for IP messages at the network layer of the Internet [6, 4, 5]. Two major protocols of IPsec are the Authentication Header (AH) [4], which provides authentication and integrity protection, and the Encapsulating Security Payload (ESP) [5], which provides encryption as well as (optional) authentication and integrity protection of IP payloads.

IPsec offers a number of advantages over other protocols being used or proposed for Internet security. Since it operates at the network layer, IPsec can be used to secure any protocol that can be encapsulated in IP, without any additional requirements. Moreover, IPsec can also be used to secure non-IP networks, such as Frame Relay, since operation of many parts of IPsec (e.g., ESP) do not necessarily require encapsulation in IP.

IV attacks are a security risk of the CBC encryption mode of block ciphers [8] which can be applied to IPsec [1]. This class of attacks makes use of an unauthenticated IV in CBC encryption, so that the attacker can do controllable changes on the first block of the decrypted plaintext by modifying the IV. We discuss IV attacks in more detail in Section 2.

When encryption is done at an intermediate layer of the protocol stack (e.g. IPsec is at the network layer) the first block modified by an IV attack includes parts of the upper-layer protocol header. In those cases, an attacker can do extraordinary things (like obtaining the whole decrypted message, as we show in Section 3) by just modifying the IV. The specifics of the results that can be obtained by such attacks depend on the upper-layer protocol in use.

In this paper, we analyze the security risks posed by IV attacks on IPsec with several possible upper-layer protocols, including TCP, UDP, (tunneled) IPv4 and v6, and L2TP. In our analysis, we consider two common block sizes: 64-bit block size, which is the block size of many popular encryption algorithms, including DES, IDEA, and Blowfish; and 128-bit block size, which will be the block size of the Advanced Encryption Standard (AES) [7]. Analysis results show that IPsec encryption can be very vulnerable to IV attacks if used carelessly. We also discuss the defense methods to protect IPsec encryption against IV attacks.

The rest of this paper is organized as follows: We describe the IV attacks and CBC encryption in Section 2. In Section 3, we analyze the impact of IV attacks on a number of protocols that are typically encapsulated in IP. In Section 4, we discuss possible defenses against the IV attacks.

2. IV Attacks against CBC encryption

In the *Cipher Block Chaining* (CBC) encryption mode of block ciphers, the data stream to be encrypted is divided into plaintext blocks $P_1, P_2, \ldots$, and each plaintext block is encrypted as

$$C_i = f_K(P_i \oplus C_{i-1}),$$

and decrypted as

$$P_i = f_K^{-1}(C_i) \oplus C_{i-1},$$

where f_K and f_K^{-1} denote the block cipher encryption and decryption, and "$\oplus$" denotes the bit-wise exclusive-or. For the special case of the first block, where there is no ciphertext C_0 to xor, an *initialization vector* (IV) is used instead:

$$\begin{aligned} C_1 &= f_K(P_1 \oplus IV) \\ P_1 &= f_K^{-1}(C_1) \oplus IV \end{aligned} \tag{1}$$

IVs are usually chosen randomly by the sender, and sent to the receiver along with (or before) the encrypted message. If an IV is sent in clear and is not protected by an authentication mechanism, an attacker can modify the IV, and therefore, can modify the first block of the decrypted plaintext according to Equation (1). We refer to such attacks, which are based on modifying the value of the IV, as the *IV attacks*. If encryption is done at an intermediate layer of the protocol stack, then the first plaintext block includes parts of the the upper-layer protocol header, and the results of an IV attack can be much more serious than just modifying a single block of plaintext. In certain cases, the attacker can obtain the whole decrypted text, as in the examples we give on IPsec in Section 3.

IV attacks can be a serious problem for IPsec encryption. IPsec has many possible configurations that enable IV attacks. First, all the IPsec encryption algorithms proposed so far use a block cipher in CBC mode. Moreover, almost all of these algorithms allow the use of cleartext unauthenticated IVs. Also in the ESP RFC [5], it is mentioned that the IV "may be carried explicitly in the Payload field" and "usually is not encrypted per se". Although [5] recommends always using authentication whenever encryption is used, which would prevent the IV attacks, authentication with ESP encryption is neither mandatory nor is it the default mode. Therefore, in practice, IPsec encryption can be very vulnerable to IV attacks. The exact results an attacker can obtain will be dependent on the protocol encapsulated in IPsec.

3. IV Attacks against IPsec

In this section we examine how IV attacks can be used against protocols encapsulated by IPsec. The protocols we examine are TCP, UDP, IPv4, IPv6, and L2TP. For the encryption block size, we consider two common sizes: 64-bit block size, which is the block size of many popular encryption algorithms, including DES, IDEA, and Blowfish; and 128-bit block size, which will be the block size of the AES [7]. Throughout the discussion, we assume that the attacker is capable of intercepting and modifying packets during transmission over the network.

The attacks in this section show that the impact of IV attacks on IPsec can be extremely serious. Our attacks include several examples where the attacker can obtain the whole decrypted message, totally defeating the encryption.

A possible complication for IV attacks comes from the upper layer protocol (e.g., TCP, UDP) checksums, which detect changes made to those protocol headers. However, these checksums are computed as a 16-bit one's complement sum, which is not one-way and can be fixed quite easily. What is needed to fix the checksum is a 16-bit scratch field that can be modified without affecting the reception of the packet, which will be used to compensate for the changes made to the fields under attack. The difference that is "added" to the checksum by the changes on the attacked fields can be "subtracted" by changing the scratch field accordingly. The method is relatively straightforward and we leave out the details.

Another complication for the IV attacks is that, in order to be able to change a header field in a desired way, first of all the original value of the field should be predictable. The fields we consider for an attack, as shown in [2], are indeed mostly predictable.

3.1. TCP

If the packet encapsulated in IPsec is a TCP packet, and a 64-bit block cipher is used to encrypt it, the fields of the TCP header that fall into the first encryption block are the Source Port, Destination Port, and Sequence Number fields. If a 128-bit block cipher is used, the fields in the first block also include the ACK Number, Data Offset, and Window Size fields, and the TCP flags.

The most significant IV attack on these fields is an attack on the Destination Port. If the attacker has a login on the same machine as the legitimate receiver, by changing the destination port number of a TCP packet, the attacker can have the packet delivered to a port of his own after the IPsec decryption. In this way, the attacker obtains the decrypted message, totally defeating the IPsec encryption. In this attack, the changes made to the TCP header do not affect the correct decryption of the packet. The decryption key is identified by the fields in the ESP header and does not depend on the fields in the inner TCP header.

Other useful attacks on TCP can be obtained by modifying the Sequence Number and the Window Size fields. By modifying the Sequence Number field, the attacker can change the ordering of the packets, which may have significant implications depending on the content of the packets. By modifying the Window Size field, the attacker can change the perceived buffer size of the connection, which may lead to the flooding or stalling of the connection.

Another possibility for an attack is to modify the Data Offset field. The Data Offset field in TCP indicates how long the TCP header length is. Modifying this field will give the attacker the ability to make part of the payload of the packet appear as a TCP option. Although not very likely, obtaining meaningful results by this kind of attack may be possible, depending on the contents of the TCP header and the payload.

The other fields in the TCP header are not likely to be useful for an attack. They can be better used to fix the checksum to compensate for the changes made to the more useful fields.

3.2. UDP

With a 64-bit block cipher, the first block of an encrypted UDP packet will consist of the whole UDP header, including the Source Port, Destination Port, Length and Checksum fields. If a 128-bit block cipher is used, the first block will additionally include the first 64 bits of the data payload. An important advantage the attacker has with UDP, as compared to TCP, is that the Checksum field is in the direct control of the attacker.

Among the fields in the first block, an attack on the Destination Port has the most significant effects. If the attacker has a login to the victim's machine, he can obtain the decrypted message by modifying the Destination Port field, similar to the attack on TCP.

Another field which can be attacked with significant impacts is the Length field, which indicates the length of the UDP packet. By decreasing the Length value, the meaning of the packet can be changed significantly; for example, "Pay Alice $10,000" may become "Pay Alice $1".

With a 128-bit block cipher, the first 64 bits of the payload is also included in the first block. So, the attacker can modify the first 64 bits of data by an IV attack. Clearly this is a very troubling prospect. The meaning of a message can be significantly altered by changing the first 64 bits. Also, the attacker conceivably has control over the *Maximum Transmit Unit* (MTU) negotiation at the lower levels of the network. If the attacker sets the MTU so that the packets will have no more than 64 bits of data, then he will have the ability to control all the data over the lifetime of the connection.

3.3. IPv4

Besides securing higher-layer protocols such as TCP and UDP, IPsec can also be used to tunnel IP packets. In case of tunneled IPv4, the fields that would be included in the first block with a 64-bit cipher are the Version, Header Length, Type of Service, Total Length, Identification, Flags, and Fragment Offset. With a 128-bit cipher, the Time to Live (TTL), Protocol, Checksum, and Source Address fields are also accessible to the attacker.

An attack on the Total Length field of IPv4 is analogous to the attack on the UDP Length field; i.e., an attacker can truncate a packet and change its meaning by changing this field.

A novel attack on IPv4 is possible by modifying the Protocol field. If the attacker has a login to the same machine as the legitimate receiver, and gets the same port number for a protocol, say UDP, as the receiver's port number for another protocol, say TCP; then by changing the Protocol field from TCP to UDP, the attacker will get the decrypted packets delivered to his UDP port, and the IPsec encryption will be totally defeated.

Another possibility for an attack is to modify the Identification and Fragment Offset fields. These fields are used to identify the location of the parts in a fragmented packet. An attack on the Fragment Offset field can change the ordering of the parts in a reassambled packet. An attack on the Identification field can swap parts between packets. Results of such attacks can be significant depending on the content of the packets.

The Flag bits can also be used to attack fragmented IP packets. If the flags are changed from a fragment setting to a whole packet setting, the receiver will accept the modified packet not as a fragment, but as an entire packet. Use of this packet for an attack will be dependent on the data contained in the packet. An attacker may also set the Last Fragment flag indicating that the fragment in question is the last fragment. This will have the effect of shortening the reassembled packet. However, in the case of transport layer being TCP or UDP, these attacks are not very useful since TCP and UDP have upper layer Length and Checksum fields.

Attacks on other fields of the IPv4 header are possible, but they are not likely to produce useful results for the attacker. These fields can be better used for checksum fixing.

3.4. IPv6

In case of tunneled IPv6, the fields accessible to the attacker with a 64-bit cipher include Version, Priority, Flow Label, Payload Length, Next Header, and Hop Limit. With a 128-bit cipher, the first 64 bits of the Source Address are also accessible.

The Next Header field replaces the Protocol field of IPv4; and an attack on this field can yield the decrypted message to the attacker, similar to the attack on the Protocol field of IPv4. Attacks on other fields are not likely to produce much useful results.

Note that IPv6 has no checksum and, therefore, modifications to the IPv6 header can be done at will.

3.5. L2TP

The Layer Two Tunneling Protocol (L2TP) is a protocol to tunnel PPP connections over wide-area networks and is used with IPsec to establish *virtual private networks* over the Internet. L2TP data packets are typically encapsulated in UDP. As we have seen from the section on UDP, a 128-bit block size will make IV attacks possible on the first 64 bits of the UDP payload, which in this case will include the Length, Tunnel ID, and Call ID fields of the L2TP header.

The most significant attack on these fields is obtained by modifying the Call ID field. In L2TP, many calls (users) can be connected to a location through the same tunnel, where each call is identified by the Call ID number. If the attacker has access to the same tunnel as the victim, he can get the decrypted packets routed to his connection at the endpoint of the tunnel by modifying the Call ID field, in a spirit similar to the attack on TCP port numbers. This attack is likely to be effective in practice since all traffic in a tunnel usually uses the same IPsec Security Association and the same key.

Another possibility for an attack is to modify the Length field. The Length field in L2TP indicates the length of the L2TP header. An attack on this field will result in a change in the perceived length of the L2TP header. Although not very likely, obtaining meaningful results by this attack may be possible, depending on the contents of the L2TP header and the payload.

An attack on the Tunnel ID field is not likely to produce any useful results, since each tunnel typically has a different encryption key. Therefore, changing the Tunnel ID will probably cause the packet to be decrypted as garbage.

4. Prevention of IV attacks

Modification of the IV in a controllable way is central to all attacks outlined in this paper. The attacker can undetectably modify the IV because the IV is sent in the clear and is not protected by cryptographic authentication.

Controllable modification of the IV can be prevented by sending the IV encrypted in the ECB mode or by sending a value that is hashed via a one-way hash function into the actual IV. Or, alternatively, modification of the IV can be totally prevented by using a constant agreed-upon IV, such as 0. Then the first block will receive only ECB protection but this is better than having the first block be modifiable.

Although there are many ways to prevent an IV attack, probably the best protection method is to always use authentication whenever encryption is used. Many reasons for authenticating secret data are present in the literature for reasons besides IV attack prevention. For example, in the case of IPsec, several attacks are known which are due to use of unauthenticated encryption [1]. Typically authentication is computationally less expensive than encryption, which also makes this solution practical. Therefore, our recommendation is to never use encryption without authentication in IPsec.

5. Conclusions

We showed that IV attacks can be a serious threat for IPsec, if IPsec is not used carefully. In the absence of authentication, an attacker may be able to totally defeat the IPsec encryption by the techniques we described in Section 3. Although there are many ways to prevent these attacks, we recommend always using authentication whenever encryption is used. Our conclusion is also supported by other attacks on IPsec, most importantly those discovered by Bellovin [1], which also make use of the fact that IPsec permits encryption to be used without authentication.

An ordinary IPsec user will not be aware of the attacks we described in this paper or the attacks Bellovin described in [1]; nor will he be aware of the recommendations in the IPsec RFCs [6, 5] to always use authentication whenever encryption is used. So, in practice, IPsec in its current form can be very vulnerable to these attacks. To prevent such failures, we recommend making authentication a mandatory feature of an IPsec operation.

Acknowledgments

We would like to thank Gargi Banerjee, Andrea Colegrove and Kim Morgenstern for many helpful discussions and suggestions.

References

[1] S. Bellovin. Problem areas for the IP security protocols. In *Sixth Usenix Unix Security Symposium*, pages 1–16, 1996.

[2] S. Bellovin. Probable plaintext cryptanalysis of the IP security protocols. In *Symposium on Network and Distributed System Security*, pages 155–160, 1997.

[3] D. Comer. *Internetworking With TCP/IP, Volume I: Principles, Protocols, and Architecture*. Prentice Hall, 1995.

[4] S. Kent and R. Atkinson. IP Authentication Header, November 1998. Internet RFC 2402.

[5] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP), November 1998. Internet RFC 2406.

[6] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol, November 1998. Internet RFC 2401.

[7] National Institute of Standards and Technology. *AES: A Crypto Algorithm for the Twenty-first Century...* http://www.nist.gov/aes.

[8] V. Voydock and S. Kent. Security mechanisms in high-level network protocols. *ACM Computing Surveys*, 15(2), June 1983.