Security flaws in tinc

Jerome Etienne jme@off.net

Abstract

This text describes security flaws in Tinc. It includes a description of the security (see section 1) and lists the possible attacks (see section 2). An attacker can modify packets, replay them and learn patterns of the plain text.

1 Security description

This section describes how tinc secures forwarded packets. The outgoing packet begins with an 'salt' of 2 bytes containing a cryptographically strong random value. It plays the role of an IV according to the manual "2 bytes of salt (random data) are added in front of the actual VPN packet, so that two VPN packets with (almost) the same content do not seem to be the same for eavesdroppers." The forwarded packet is appended. The couple salt and forwarded is padded to be 64bit aligned (blowfish's block size). The whole (salt, forwarded packet and padding) is encrypted with blowfish in CBC.

2 Vulnerabilities

This section explains how an attacker can modify packets (see section 2.1) , replay them (see section 2.3), learn pattern of the plain text (see section 2.2).

2.1 No packet authentication

The aim of encryption is to make the data unreadable for anybody who doesn't know the key. It doesn't prevent an attacker from modifying the data. People assume that an attacker won't do it because the attacker wouldn't be able to choose the resulting clear text. But this section shows that the attacker can choose the resulting clear text to some extends and that modifying the cypher text data may be interesting even if the attacker ignores the result.

2.1.1 To insert random data

If the attacker modifies the cipher text without choosing the resulting clear text, it will likely produce random data. The legitimate user won't detect the modification and will use them as if they were valid. As they likely appears random, it will result of a Denial of Service (aka DoS).

2.1.2 To insert chosen data

The encryption mode is CBC[oST81, sec 5.3]. CBC allows cut/past attacks i.e. the attacker can cut encrypted data from one part of a packet and paste them in another location. As both data sections have been encrypted by the same key, the clear text won't be completely random data.

This lack of authentication isn't a CBC flaw. Authentication isn't considered a aim of the encryption mode, so most modes (e.g. ECB, CFB, OFB) doesn't authenticate the data. To use another mode would be flawed in the same way except if they explicitly protect against forgery. Recently some modes including authentication popped up to speed up the encryption / authentication couple but as far as i know they are all patented.

In very short, encrypting with CBC is Cn=Enc(Cn-1 xor Pn) where Enc(x) is encrypting x, Pn is the nth block of plain text and Cn the nth block of cipher text. For the first block, Cn-1 is an Initial vector (aka IV) which may be public and must be unique for a given key. The decryption is Pn = Dec(Cn) xor Cn-1. See [oST81, sec 5.3] for a longer description of CBC.

If the attacker copies s blocks from the location m to n (aka [Cn,...,Cn+s-1] == [Cm,...,Cm+s-1]), Pn+1 up to Pn+s-1 will the same as Pm+1 to Pm+s-1 and Pn will likely appears random. Cn (i.e. Cm) will be decrypted as Pn = Dec(Cm) xor Cn-1 but Cm-1 and Cn-1 are different so Pn will likely appears random. Nevertheless Pn+1 = Dec(Cn+1) xor Cn = Dec(Cm+1) xor Cm = Pm+1, so Pn+1=Pm+1. So if the attacker has an idea of the content of a group of blocks in a packet, he can copy them to the Nth block, thus it can choose the content of it without being detected.

As usual packets aren't designed to appears random, its content may be predictable to some extents (e.g. IP header) The attacker may use such informations to guess the contents and do a knowledgeable cut/past.

2.2 Insecure IV

The aim of an IV is to hide the repetitive patterns inside the the encrypted plain text, so it must be unique for a given key. Tinc's IV, called salt in the source, are random so they aren't guaranteed to be unique and are vulnerable to the birthday paradox. Moreover the IV is only 16bit long, so as a rule of thumb if tinc forwards 255 packets, there is a probability of 50% to have at least 2 packets with the same IV.

2.3 No anti-replay protection

Tinc doesn't include any protection against packet's replay, so an attacker who eavesdrops the encrypted packets can successfully replay them later and the destination will consider them as legitimate.

The manual section 6.3.2 claims "There is no extra provision against replay attacks or alteration of packets. However, the VPN packets, normally UDP or TCP packets themselves, contain checksums and sequence numbers. Since those checksums and sequence numbers are encrypted, they automatically become cryptographically secure. The kernel will handle any checksum errors and duplicate packets."

We believe it is risky to base the security on assumption on the forwarded packets. Moreover in this case, the assumptions are incorrect. UDP doesn't have any sequence number. TCP do have sequence numbers but, for example, an attacker can replay a TCP syn packet to perform a SYN flood attack on a server behind the tinc peer.

3 Conclusion

This text describes how an attacker can modify packets, replay them and learn patterns of the plain text. The holes are real, practical and independant. They may be combined to perform stronger attacks.

References

[oST81] National Institute of Standards and Technology. implementing and using the nbs data encryption standard. *Federal information processing standards fips74*, April 1981.