

[UNIX] Security Analysis of VTun

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-01/0042.html>

From: support@securiteam.com

Date: 01/11/02

From: support@securiteam.com

To: list@securiteam.com

Date: Fri, 11 Jan 2002 15:10:50 +0100 (CET)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

Security Analysis of VTun

SUMMARY

The following text describes security flaws in <http://vtun.sourceforge.net/> VTunD, a program that is easiest way to create Virtual Tunnels over TCP/IP networks with traffic shaping, compression and encryption. It includes a description of the security based on the source and lists the possible attacks. An attacker can modify packets, replay them, learn pattern of the plain text or easily guess low-entropy password.

DETAILS

Vulnerable systems:

VTun version 2.5b1

This text is a security analysis of VTun. It includes a description of the security (see section 'security description') based on the source and lists the possible attacks (see section 3). An attacker can modify packets, replay them, learn pattern of the plain text or easily guess low-entropy password.

Introduction

From the man page, "VTun provides the method for creating Virtual Tunnels over TCP/IP networks and allows to shape, compress, and encrypt traffic in

that tunnels." The analysis has been done on VTun version 2.5b1 which has been downloaded from <<http://vtun.sourceforge.net>>
<http://vtun.sourceforge.net>.

Security description:

The security has been analyzed from the source as the distribution does not contain any detailed description.

Packet forwarding:

The forwarded packets are encrypted with blowfish in ECB using MD5 (user password) as encryption key (see lfd_encrypt.c). As ECB requires the cipher text to be block aligned and blowfish has 64-bit blocks, the packet is 64bit aligned. The pad is zeroes pre-pended to the packet and the first byte of the packet is its length.

Connection establishment:

During the connection Establishment, the client authenticates itself to the server with a challenge/response scheme (i.e. a simple way to authenticate without sending passwords in clear) based on a user password. The challenge is 16bytes of random (see VTUN_CHAL_SIZE) chosen by the server. They are encrypted with a key equal to MD5 (user password). The server sends the encrypted challenge to the client; the client decrypts it and replies it.

The above explanation assumes the HAVE_SSL is defined. If it is not, the authentication is very insecure because the challenges is just XOR-ed with the password, and the challenge is based on rand() output which is known as easily predicable.

Vulnerabilities:

This section explains how an attacker can modify packets, replay them, learn pattern of the plain text or easily guess low-entropy password.

Forwarded packet aren't authenticated:

The forwarded packets are not authenticated, so an attacker can modify them without being detected. The aim of encryption is to make the data unreadable for anybody who does not know the key. It does not prevent an attacker from modifying the data. People assume that an attacker will not do it because the attacker would not be able to choose the resulting clear text. However, this section shows that the attacker can choose the resulting clear text to some extends and that modifying the cipher text data may be interesting even if the attacker ignores the result.

Inserting random data:

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

Inserting chosen data:

The encryption mode used by encrypted loop device is ECB[oST81]. ECB

allows cut/past attacks i.e. the attacker can cut encrypted data from one part of a packet and paste them anywhere in another packet. As both data sections have been encrypted by the same key, the clear text will not be completely random data.

This lack of authentication is not an ECB flaw. Authentication is not considered an aim of the encryption mode, so most modes (e.g. CBC, CFB, and OFB) do not 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 we know, they are all patented.

Easy dictionary attacks:

The authentication is based on a secret key chosen by the user. The key is trivially derived from the user password by computing MD5 (user password).

Unfortunately, users often choose low-entropy passwords because those are easier to remember, even if it is a bad behavior from a security point of view. This allows attackers to try dictionary attacks i.e. to try likely password (e.g. jack the ripper). This weakness isn't inherently a VTun weakness as the password choice depends on the users. He may choose a random password (e.g. /dev/random output) and will not be vulnerable.

When the security ultimately relies on a low-entropy password chosen by a user, dictionary attacks cannot be stopped but they can be made sufficiently harder to be impractical (e.g. salt, key derivation sufficiently slow). VTun does not use those tricks.

No replay protection:

VTun does not include any protection against packet replay, so an attacker who can eavesdrop on the encrypted packets can successfully replay them later as the destination will consider them as legitimate. They can be replayed inside the same tunnel or in another instance the tunnel. The attacker can even replay them to the source: a packet from A to B can be send to A that will accept it.

Usage of ECB:

VTun uses blowfish with ECB but ECB does not hide the patterns inside the plain text. A given plain text block will produce the same cipher text block independently of the packet in which it is and of its location inside them. The attacker can recognize the repetition of identical cipher text blocks and obtain information on the plain text.

Conclusion:

This text describes vulnerabilities of VTun security. An attacker can modify packets, replay them, learn pattern of the plain text or easily guess low-entropy password. All those attacks are independent and can be combined to perform even stronger attacks.

Securiteam: [UNIX] Security Analysis of VTun

References:

[oST81]

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

ADDITIONAL INFORMATION

The information has been provided by <mailto:jme@off.net> Jerome Etienne.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

-
- **Previous message:** support@securiteam.com: "[NEWS] Netscape ?wp-html-rend Denial of Service Attack"
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)