MWR LABS

Security Advisory

Tinc Authentication Bypass

24/10/2018

Software	Tinc
Affected Versions	1.0.34 and earlier
CVE Reference	CVE-2018-16737, CVE-2018-16738, CVE-2018-16758
Author	Michael Yonli
Severity	High, High, Medium, CVSS: 8.1, 8.1, 6.8
Vendor	The tinc development team
Vendor Response	Patch available - https://www.tinc-vpn.org/download/

Description:

tinc (https://www.tinc-vpn.org/) is a piece of software used to create Virtual Private Networks (VPNs).

A lack of authenticity verification enables attackers to bypass the authentication scheme, due to a decryption oracle, as well as to modify meta-messages, if they are in a position to Man in the Middle (MitM) traffic. Meta-messages are used to manage the VPN network itself, these messages may for example negotiate new keys, authenticate new nodes or change the network structure, but do not carry any of the data that is sent over the network itself as a medium.

Impact:

Attackers can remotely bypass the authentication protocol allowing them to replace other nodes with themselves (CVE-2018-16737 and CVE-2018-16738) and meta-messages may be modified in transit (CVE-2018-16758).

Cause:

The authenticity of meta-messages is never verified which results in the inability to detect maliciously modified or replayed meta-messages. In addition a decryption oracle for authentication challenges was found.

Interim Workaround:

Upgrading to versions past 1.0.30 limits the content of a meta-message that can be modified by an attacker due to a different cipher mode being used and also significantly increases the complexity of successfully bypassing the authentication scheme. Setting the ping timeout to a low value also makes a successful bypass of the authentication more difficult, as the attack has to be completed within the timeout window and involves sending a large number of messages, which took multiple seconds on a test system.

Solution:

Apply the vendor supplied patch for the issue available at https://www.tinc-vpn.org/download/.

Technical details

The meta-messages prior to version 1.0.30 uses the Blowfish cipher in Output Feedback (OFB) mode without a Message Authentication Code (MAC) which allows an attacker to tamper with intercepted data.

Authentication consists of decrypting a challenge and responding with its hash. This challenge in its plaintext form is a hex string. The protocol verifies that the challenge is a valid hex string and closes the connection upon failure.

```
if(!hex2bin(buffer, c->mychallenge, len)) {
    logger(LOG_ERR, "Got bad %s from %s(%s): %s", "CHALLENGE", c->name, c->hostname,
"invalid challenge");
    return false;
}
```

An attacker can replay a captured challenge while flipping bits in order to infer whether flipping these bits on the plaintext would result in an invalid hex string. This can be used to completely decrypt the challenge and bypass the authentication.

Versions after and including 1.0.30 use the Advanced Encryption Standard (AES) cipher in Cipher Feedback (CFB) mode which limits an attacker to modifying the bytes that map to the last ciphertext block of a message. Decrypting the challenge becomes more difficult due to having to rely on timing as a means of reading the oracle rather than the state of the connection.

Detailed Timeline

Date	Summary
2018-09-06	Issue reported to vendor
2018-09-09	CVE-2018-16737, CVE-2018-16738 and CVE-2018-16758 assigned
2018-10-08	Patch released
2018-10-24	Advisory published