XEP-0344: Impact of TLS and DNSSEC on Dialback

Philipp Hancke
mailto:fippo@andyet.com
xmpp:fippo@goodadvice.pages.de

Dave Cridland
mailto:dave.cridland@surevine.com
xmpp:dave.cridland@surevine.com

2015-03-23
Version 0.3

Status	Type	Short Name
Experimental	Standards Track	N/A

This specification provides documentation how Server Dialback is used together with Transport Layer Security, and discusses how the security considerations of Dialback are changed by the introduction of TLS and/or DNSSEC.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction **1**

2 Protocol **1**

2.1 Dramatis Personae 1

2.2 Classic Dialback Flow 1

2.3 XMPP Exchanges in Classic Dialback over TLS 2

2.4 Dialback without dialback flow 4

2.5 XMPP Exchanges in Dialback without dialback 5

2.6 Same Certificate shortcut 7

2.7 XMPP Exchanges in Same Certifiate shortcut 8

3 Security Considerations **9**

3.1 Dialback without dialback shortcut 9

3.2 Same Certificate shortcut 10

3.3 DNSSEC 10

4 IANA Considerations **10**

5 XMPP Registrar Considerations **10**

1 Introduction

Although Server Dialback (XEP-0220) [1] describes dialback as being run before any other negotiation, it is typically run over TLS where supported. This allows it to be used as a simple convenient fallback to X.509 Strong Authentication within the TLS layer, as described in RFC 6120 [2], and also affords greater protection to the exchange.
This document describes these practises, and also describes various functionally equivalent shortcuts to the protocol, including that known as "dialback without dialback".

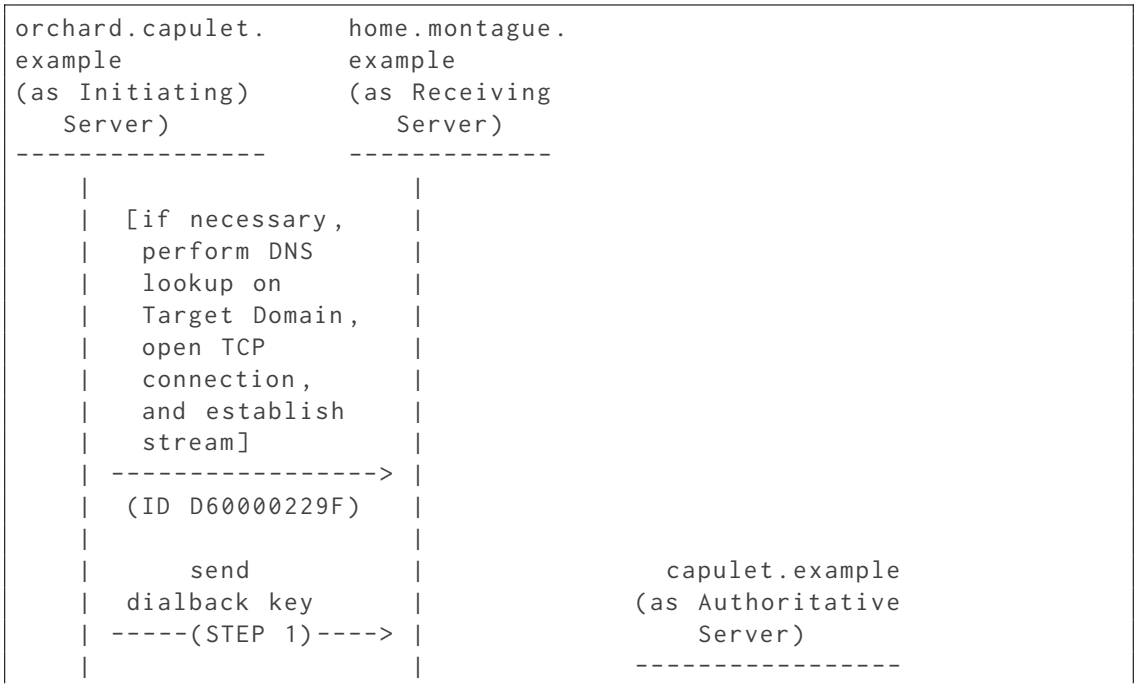
2 Protocol

2.1 Dramatis Personae

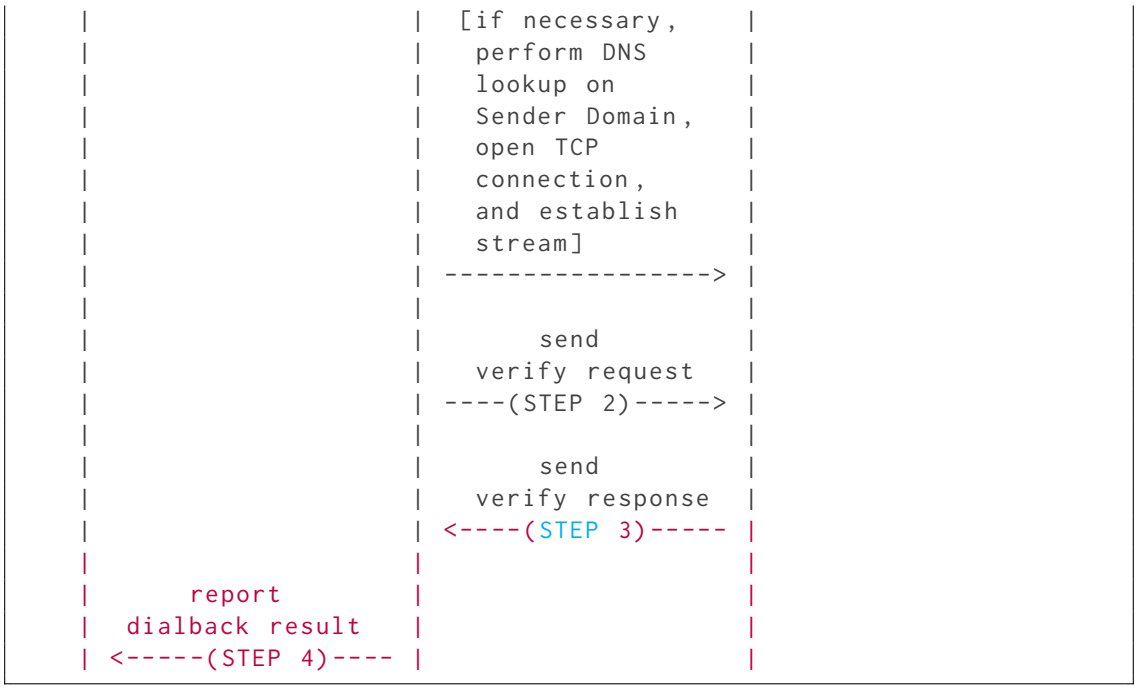
This document will tell a tale of two servers; orchard.capulet.example is trying to contact home.montague.example. Each server operates a single domain; these are capulet.example and montague.example respectively.

2.2 Classic Dialback Flow

The traditional pattern is shown here:



[1] XEP-0220: Server Dialback <https://xmpp.org/extensions/xep-0220.html>.
[2] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.



2.3 XMPP Exchanges in Classic Dialback over TLS

This traditional pattern involves the following protocol exchanges when dialback over TLS is used:

Listing 1: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 2: Receiving Server Responds with a stream header and advertises TLS feature

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
    <required/>
```

```
  </starttls>
</stream:features>
```

Listing 3: Initiating Server Sends STARTTLS command

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 4: Receiving Server informs Initiating Server to proceed

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 5: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 6: Receiving Server Responds with a stream header

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  </mechanisms>
</stream:features>
```

Listing 7: Initiating Server Sends Dialback Key (Step 1)

```
<db:result
    from='capulet.example'
    to='montague.example'>
  b4835385f37fe2895af6c196b59097b16862406db80559900d96bf6fa7d23df3
</db:result>
```

The Receiving Server may need to establish a connection to the Authoritative Server at this point.

Listing 8: Receiving Server Sends Verification Request to Authoritative Server (Step 2)

```
<db:verify
    from='montague.example'
```

```
    id='D60000229F'
    to='capulet.example'>
  b4835385f37fe2895af6c196b59097b16862406db80559900d96bf6fa7d23df3
</db:verify>
```

Listing 9: Receiving Server is Informed by Authoritative Server that Key is Valid (Step 3)

```
<db:verify
  from='capulet.example'
  id='D60000229F'
  to='montague.example'
  type='valid'/>
```

Listing 10: Initiating Server Receives Valid Verification Result from Receiving Server (Step 4)

```
<db:result
  from='montague.example'
  to='capulet.example'
  type='valid'/>
```

2.4 Dialback without dialback flow

If during the initial connection, home.montague.example is able to determine that the certificate presented is trustworthy, and authenticates orchard.capulet.example as authorized to offer the XMPP service for capulet.example, then the flow can be shortcutted heavily, allowing the entire Authoritative Server process to be elided.
This is particularly useful in cases where the dialback exchange is a subsequent exchange used in piggybacking, as it remains the only solution for piggybacking with strong authentication.

```
orchard.capulet.     home.montague.
example              example
(as Initiating)      (as Receiving
   Server)              Server)
----------------     -------------
    |                      |
    |  [if necessary,      |
    |   perform DNS        |
    |   lookup on          |
    |   Target Domain,     |
    |   open TCP           |
    |   connection,        |
    |   and establish      |
    |   stream]            |
    | ----------------->   |
    |  (ID D60000229F)     |
    |                      |
    |      send            |
```

```
    |  dialback key      |
    | -----(STEP 1)----> |
    |                    |
    |                    |  [observe certificate
    |                    |   trustworthy and
    |                    |   correct for capulet.example
    |                    |   as per RFC 6125]
    |                    |
    |      report        |
    |  dialback result   |
    | <-----(STEP 4)---- |
```

2.5 XMPP Exchanges in Dialback without dialback

This traditional pattern involves the following protocol exchanges when dialback over TLS is used:

Listing 11: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 12: Receiving Server Responds with a stream header and advertises TLS feature

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
    <required/>
  </starttls>
</stream:features>
```

Listing 13: Initiating Server Sends STARTTLS command

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 14: Receiving Server informs Initiating Server to proceed

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 15: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 16: Receiving Server Responds with a stream header

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism name='EXTERNAL'/>
  </mechanisms>
</stream:features>
```

Note that having authenticated the certificate and found it authorized for capulet.example, montague.example has offered EXTERNAL above. It's not clear why capulet.example does not avail itself of the offer below; however it should be noted that EXTERNAL would not be available with piggybacking for example.

Listing 17: Initiating Server Sends Dialback Key (Step 1)

```
<db:result
    from='capulet.example'
    to='montague.example'>
  b4835385f37fe2895af6c196b59097b16862406db80559900d96bf6fa7d23df3
</db:result>
```

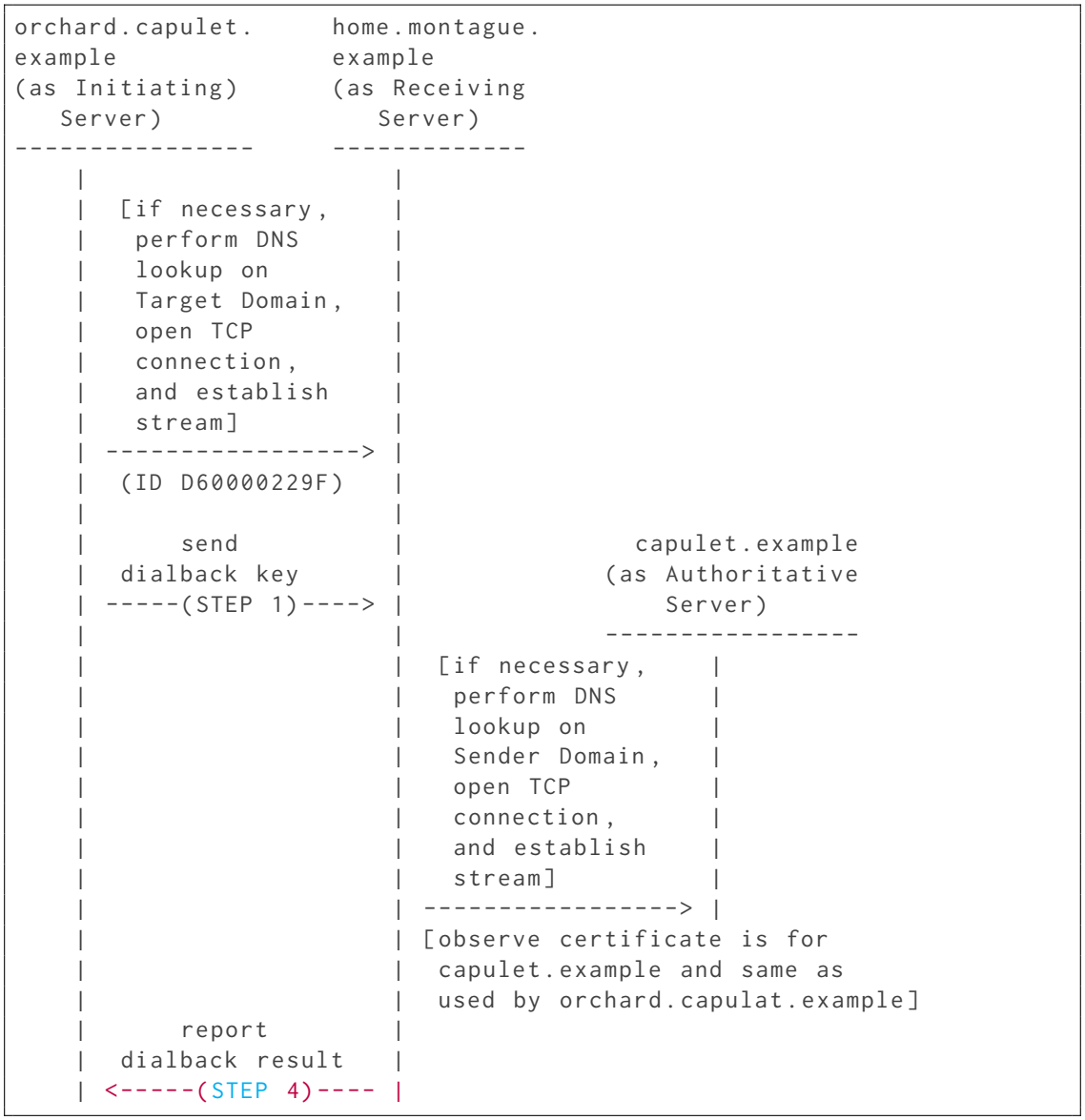
Critically, it is at this point that home.montague.example both authenticates and checks authorization on the certificate, or at least ensure that the certificate presented at this stage matches that presented at the initial handshake.

Listing 18: Initiating Server Receives Valid Verification Result from Receiving Server (Step 4)

```
<db:result
  from='montague.example'
  to='capulet.example'
  type='valid'/>
  from='montague.example'
  to='capulet.example'
  type='valid'/>
```

2.6 Same Certificate shortcut

If during the initial connection, the Receiving Server is unable to determine that the certificate presented is trustworthy but the Authoritative Server presents the same certificate as the Originating Server, the <db:verify/> step can be elided.
Note: the Receiving Server MUST still check that the hostname in the certificate matches.
Essentially, this replaces the Dialback Key Validation step from Dialback Key Generation and Validation (XEP-0185) [3] with the somewhat more elaborate proof of posession of the private key associated with the certificate.



[3] XEP-0185: Dialback Key Generation and Validation <https://xmpp.org/extensions/xep-0185.html>.

2.7 XMPP Exchanges in Same Certifiate shortcut

This pattern involves the following protocol exchanges:

Listing 19: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 20: Receiving Server Responds with a stream header and advertises TLS feature

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
    <required/>
  </starttls>
</stream:features>
```

Listing 21: Initiating Server Sends STARTTLS command

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 22: Receiving Server informs Initiating Server to proceed

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Listing 23: Initiating Server Opens Stream

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='capulet.example'
        to='montague.example'
        version='1.0'>
```

Listing 24: Receiving Server Responds with a stream header

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
```

```
        id='D60000229F'
        from='montague.example'
        to='capulet.example'
        version='1.0'>
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  </mechanisms>
</stream:features>
```

Listing 25: Initiating Server Sends Dialback Key (Step 1)

```
<db:result
    from='capulet.example'
    to='montague.example'>
  b4835385f37fe2895af6c196b59097b16862406db80559900d96bf6fa7d23df3
</db:result>
```

The Receiving Server may need to establish a connection to the Authoritative Server at this point. Here we assume that this connection is using TLS and the certificate presented by the Authoritative Server is the same as the one used by the Originating Server and contains the domain name claimed by the Originating Server.

Listing 26: Initiating Server Receives Valid Verification Result from Receiving Server (Step 4)

```
<db:result
  from='montague.example'
  to='capulet.example'
  type='valid'/>
```

3 Security Considerations

With respect to **XEP-0220**'s security considerations, the adaptations in this document add at minimum channel encryption and integrity, which forces an attacker into making an active attack, rather than passive eavesdropping. This raises the cost of an attack significantly. However, unless the certificates are authenticated, there is still a man-in-the-middle attack possible, and the reliance on unauthenticated DNS remains problematic.

3.1 Dialback without dialback shortcut

Use of the "Dialback without dialback" shortcut described in section 2.4 raises the level of authentication to that of the TLS/SASL-EXTERNAL process described in **RFC 6120**, and is thought to be indistinguishable from a security standpoint. As such, the security considerations relating to this in **RFC 6120** et al apply.

3.2 Same Certificate shortcut

Use of the ”Same Certificate” shortcut described in section 2.6 is not thought to materially alter the security profile beyond that described above. In particular, it does not alter the level of trust an implementation may put in authentication.

3.3 DNSSEC

If both SRV and A/AAAA records are protected by DNSSEC, this means that the correct address for the peer can be proven, removing DNS forgery as an attack vector. Without TLS, it is however still possible to mount an array of attacks, including IP spoofing and eavesdropping. With TLS, however, the situation improves. Since TLS protects against a naïve IP spoofing attack, a routing protocol attack (such as BGP hijacking) is required to forge the server.
In addition, it is of critical importance to check the certificate at the time when the dialback result is received, and not only in the initial handshake. This protects against an attack based around renegotiation.

4 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [4].

5 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar.

[4] The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.