IQ Spoofing

Thijs Alkemade - IETF 89 London

March 4, 2014

History

> February 25, 2013, on security@pidgin.im:
>
> “libpurple does not check whether the ‘from’ attribute on an IQ reply to a vCard request is valid, allowing the attacker to cause a NULL pointer dereference in the vCard handler.”

History

Target:

```
<iq type=’get’ id=’purple5f8085f8’
    to=’attacker@jabber.org’>
    <vCard xmlns=’vcard-temp’/>
</iq>
```

Attacker:

```
<iq type="result" id="purple5f8085f8" from="@"
    to="user@jabber.org/dbe143fe">
    <vCard xmlns="vcard-temp">
        <NICKNAME>X</NICKNAME>
    </vCard>
</iq>
```

Callbacks

Register a callback to be called when a reply comes in:

```
jabber_iq_set_callback(JabberIq *iq,
                       JabberIqCallback *callback,
                       gpointer data);
```

Callbacks

Register a callback to be called when a reply comes in:

```
jabber_iq_set_callback(JabberIq *iq,
                       JabberIqCallback *callback,
                       gpointer data);
```

The ‘id’ matches the original IQ’s ‘id’ ⇒ it’s a reply!

Callbacks

Register a callback to be called when a reply comes in:

```
jabber_iq_set_callback(JabberIq *iq,
                       JabberIqCallback *callback,
                       gpointer data);
```

The ‘id’ matches the original IQ’s ‘id’ ⇒ it’s a reply!
Any attacker that knows the ‘id’ can spoof a reply: vCards, rosters, etc.

Servers

January 2014: Fixed, many servers don't work anymore. Servers are sending replies "on behalf of the user's own account" with 'from' address:

- `user@example.com/resource` (allowed in RFC3920, but not in 6120)
- `example.com` (not allowed)

Other vulnerable libraries

- Go xmpp package (#13) – Fixed
- Messages.app (#16147049) – No acknowledgement
- Miranda (#569) – No acknowledgement
- SleekXMPP (#278) – Fixed
- Smack (#533) – Acknowledged, not fixed
- Strophe.js (#56) – Fixed
- XMPPFramework (#300) – Fixed

Prevention

Why are so many people doing this wrong?

Adam Langley, Go xmpp package:

"Clearly, here, I misunderstood the id to be a hop-to-hop id not end-to-end."

Prevention

RFC 6120, 8.1.3:

The ‘id’ attribute is used by the originating entity to track any response or error stanza that it might receive in relation to the generated stanza from another entity (such as an intermediate server or the intended recipient).
It is up to the originating entity whether the value of the ‘id’ attribute is unique only within its current stream or unique globally.

Prevention

draft-alkemade-xmpp-iq-validation goals:

- Emphasize these are end-to-end
- Emphasize clients are responsible for verification
- Maybe: Give recommendations about picking good ‘id’ attributes

ids

What should the recommendations for IQ ids be?

- Counters leak presence information.
    - Counters reveal number of stanzas sent
    - Correlate that over groups of people to find who is talking to whom.

ids

What should the recommendations for IQ ids be?

- Counters leak presence information.
    - Counters reveal number of stanzas sent
    - Correlate that over groups of people to find who is talking to whom.
- Predictable ids can still be abused (in situations like MUCs).

ids

What should the recommendations for IQ ids be?

- Counters leak presence information.
  - Counters reveal number of stanzas sent
  - Correlate that over groups of people to find who is talking to whom.
- Predictable ids can still be abused (in situations like MUCs).
- Uniqueness?
  - Clients can easily check whether an ‘id’ is currently in use already
  - Do we really need uniqueness in a stream?

ids

My suggestion:

- Generate a 64-bit number from a cryptographically-secure random number generator.
- If a handler for that ‘id’ exists: try again.

More issues

- Not verifying the source of roster pushes (Smack, XMPPFramework, Messages.app).
- Servers allowing messages with malformed/spoofed ‘from’ JID’s to go through. Not yet investigated...