The Internet Engineering Task Force (IETF)

The goal of the IETF is to make the Internet work better.

"This is the mailing list for the Transport Layer Security working group of the IETF."

[TLS] MITM attack on delayed TLS-client auth through renegotiation

---

- *To*: tls at ietf.org
- *Subject*: [TLS] MITM attack on delayed TLS-client auth through renegotiation
- *From*: Martin Rex <Martin.Rex at sap.com>
- *Date*: Wed, 4 Nov 2009 18:28:00 +0100 (MET)
- *Delivered-to*: tls at core3.amsl.com
- *List-archive*: <http://www.ietf.org/mail-archive/web/tls>
- *List-help*: <mailto:tls-request@ietf.org?subject=help>
- *List-id*: "This is the mailing list for the Transport Layer Security working group of the IETF." <tls.ietf.org>
- *List-post*: <mailto:tls@ietf.org>
- *List-subscribe*: <https://www.ietf.org/mailman/listinfo/tls>, <mailto:tls-request@ietf.org?subject=subscribe>
- *List-unsubscribe*: <https://www.ietf.org/mailman/listinfo/tls>, <mailto:tls-request@ietf.org?subject=unsubscribe>
- *Reply-to*: mrex at sap.com

```
After elaborating so much about the client cert authentication
through renegotiation with Microsoft IIS, I'm beginning to believe
that there is a potential security problem with that scheme,
because it is susceptible to a MITM attack.

How serious the problem is depends on whether and how the client
performs the server identification on the renegotiation TLS handshake.

The problem:  when Microsoft IIS is configured to request a client
certificate after having received the request, then it WILL perform
an unauthenticated request!  Sending the reply back only to the
authenticated client is a poor excuse for acting on an unauthenticated
request.


Attack scenario:


             sess1                                    sess2
  TLS client <--->  rogue TLS server (doing MITM)  <--->  victim MS IIS
```

The rogue TLS server waits for innocent clients to connect and offers
to accept the same TLS client certs than the victim MS IIS server.

If a TLS client with a promising TLS client cert connects (sess1),
then the rogue TLS server establishes an anonymous TLS connection
(sess2) with the victim MS IIS server and sends the request it wants
performed (URL with command parameters) to the victim MS IIS server--
which replys with a Hello Request asking for a full TLS handshake
of an entirely new and independent authenticated TLS session (sess3)

At this point, the rogue TLS server starts relaying the TLS
handshake messages between the TLS client and the victim MS IIS,
i.e. it forwards all handshake messages it receives over sess2
to the client over sess1, and likewise forward all handshake
messages it receives over sess1 to the victim MS IIS over sess2.

The ChangeCipherSpec is the last Handshake message on each
direction that is decrypted/encrypted under the original sess1/sess2
settings, for all further communication, the rogue TLS server (MITM)
will forward the incoming network data 1:1 to the other side
(because that is protected under keys known only to the TLS client
and to the victim MS IIS.

The victim MS IIS server has no means to detect that it has been
attacked. Whether the TLS client cares about receiving a different
TLS server cert in the renegotiation handshake and what it will
do about it -- if anything, depends entirely on the application
of the TLS client. At the TLS level, everything looks just fine.

I'm not sure that all clients will repeat the server authentication.
They can be expected to verify the server certificate before sending
off their request. Depending on the API architecture, they may not
always realize or care that a renegotiation was performed while they
were sitting on SSL_read() waiting for the server reply.

And even if they decide to perform an additional server endpoint
identification, they might be doing it after the renegotiate
handshake has been successfully completed -- a point where the
victim MS IIS has started performing the action requested
in the anonymous request from the MITM.

-Martin