# TLS and SSLv3 vulnerabilities explained

DRAFT

**Thierry ZOLLER**
Principal Security Consultant
contact@g-sec.lu
**http://www.g-sec.lu**

G-SEC

G-SEC™ is a **vendor independent** Luxemburgish led security consulting group that offers IT Security consulting services on an organizational and technical level. Our work has been featured in New York Times, eWeek, ct', SAT1, Washington Post and at conferences ranging from Hack.lu to Cansecwest.

# Table of Contents

## Synopsis

Around the 09/11/2009 Marsh Ray, Steve Dispensa and Martin Rex published details[1] about a vulnerability affecting the renegotiation phase of the TLS & SSLv3 protocol. The vulnerability is being tracked under CVE-2009-3555[2] | VU#120541[3] and affects a multitude of platforms and protocols, the impact of this vulnerability varies from protocol to protocol and research into those is currently ongoing.

When speaking of a "Man in the Middle" attack, it is often assumed that data can be altered or changed. Indeed an attacker that sits in the middle of a connection (hence it's name) is often able to do so. In this particular case however the attacker piggybacks an existing authenticated and encrypted TLS sessions in order to (prefix) inject arbitrary text of its choice. The attacker may not read/alter the other TLS session between the "client" and the "server". See Chapter 3 - "Example of an attack scenario..." for more details

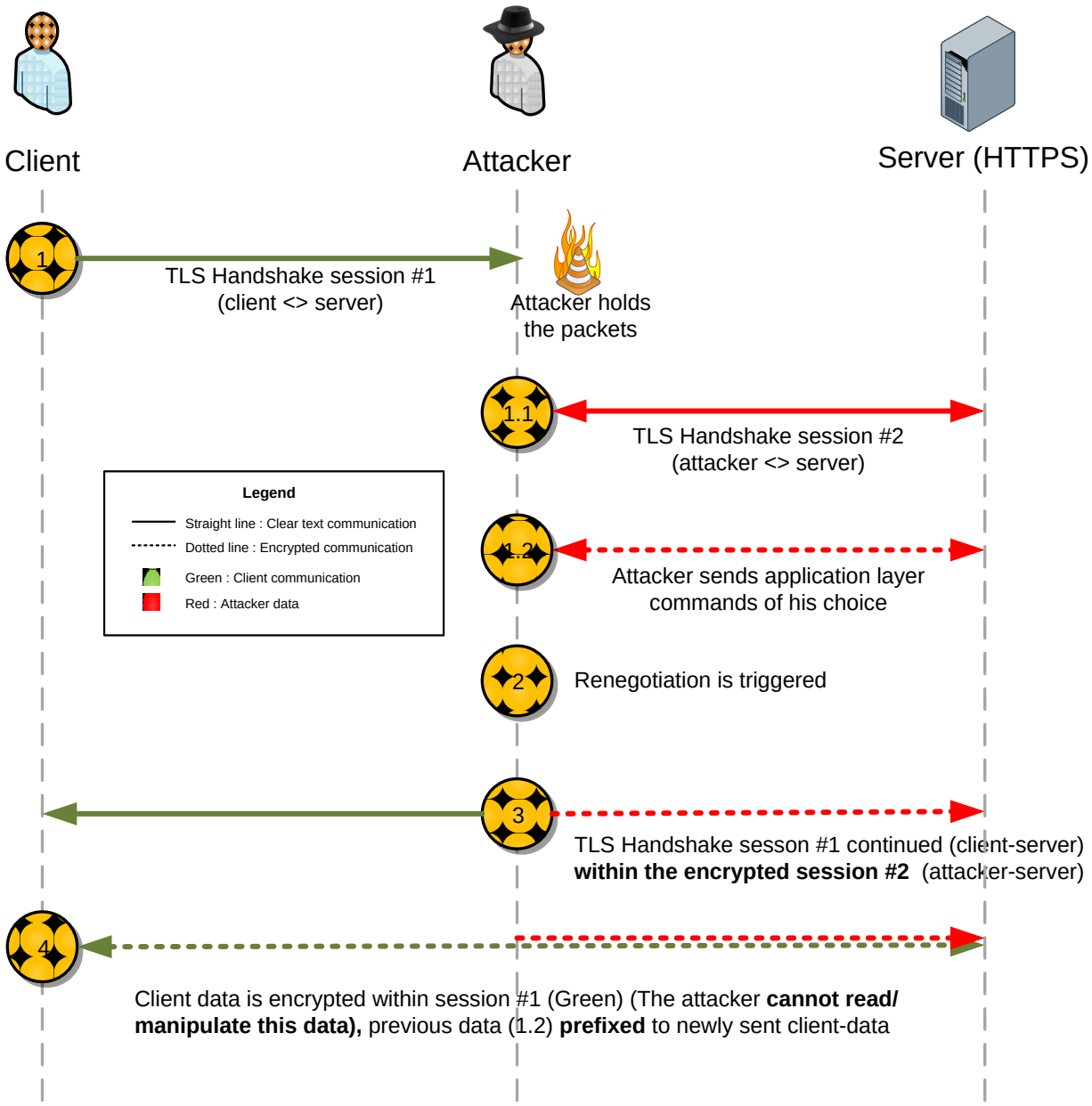
This paper explains the vulnerability for a broader audience and summarizes the information that is currently available. The document is prone to updates and is believed to be accurate by the time of writing.

## Revisions

| *Version* | *Date* | *Annotations* |
|---|---|---|
| 0.8 | 09.11.2009 | Initial draft |
| 0.81 | 10.11.2009 | Adding general and specific example |
| 0.9 | 12.11.2009 | Added vulnerability requirements, protocol overview |
| 0.91 | 12.11.2009 | Initial public draft release at http://www.g-sec.lu/ |
| 0.92 | 13.11.2009 | Corrected few errors |
| 0.93 | 17.11.2009 | Added test cases and SMTP over TLS details |

[1] http://www.extendedsubset.com/
[2] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3555
[3] http://www.kb.cert.org/vuls/id/120541

## Generic example: TLS renegotiation prefix injection vulnerability



## Details

1 "Client" starts the TLS handshake – Attacker does not forward these immediately

1.1 The attacker negotiates a new session performs a full TLS exchange

1.2 The attacker sends application level commands over the previously established TLS session (#2)

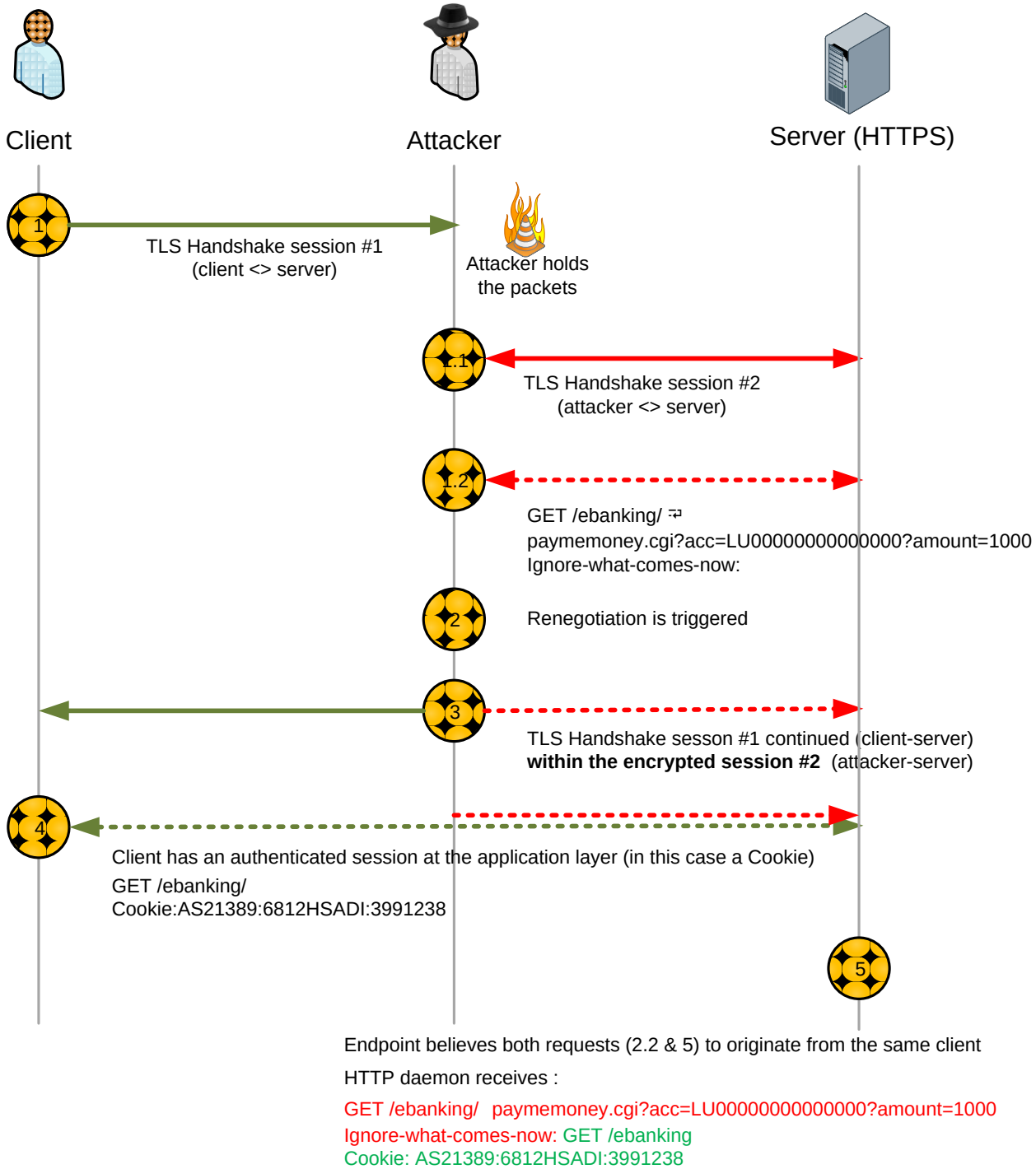
2 Renegotiation is triggered either

1. because of Certificate based auth (server sees get /dir and decides it needs an certificated for „directory")
2. due to different cipher requriements on different ressources (Server initiated)
3. by the client

3 The TLS handshake started at 1 and hold back by the attacker, is now being let to the server which performs a new TLS Handshake **over the previously established encrypted TLS session #2** (Attacker<>Server)

4 The TLS endpoint, due to the renegotiation has to take into the account the previously sent data (per spec), the endpoint believes the previous data (1.2) to have been send from the same client. As such this request is **prefixed to the one issued by the client in 4** (See HTTPS example for a more explicit example)

## Specific Example : TLS renegotiation clear stream prefix injection vulnerability abusing HTTPS



## Details

This is a simplistic example of how this vulnerability might be used to affect HTTPS (with client-cert, or without). We are aware that in this case a simple XSRF[4] attack could have achieved the same effect, however this is a easy to understand example

1.1 The attacker negotiates a new session performs a full TLS exchange

1.2 The attacker sends a GET request to a fictional weak e-banking application,

2 Renegotiation is triggered

3 The TLS handshake started at 1 and hold back by the attacker, is now being let to the server which performs a new TLS Handshake **over the previously established encrypted TLS session #2** (Attacker<>Server)

4 The TLS endpoint, due to the renegotiation has to take into the account the previously sent data (per spec), the endpoint believes the previous data (1.2) to have been send from the same client

The requests

1.2 : Attacker -> server

```
GET /ebanking/  paymemoney.cgi?acc=LU00000000000000?amount=1000
Ignore-what-comes-now:
```

And

4: Client->server

```
GET /ebanking
Cookie: AS21389:6812HSADI:3991238
```

5 The request is prefixed to the request issued by the client in (4)

Are merged into

```
GET /ebanking/  paymemoney.cgi?acc=LU00000000000000?amount=1000
Ignore-what-comes-now: GET /ebanking
Cookie: AS21389:6812HSADI:3991238
```

Interpreted by the HTTP daemon as :

```
GET /ebanking/  paymemoney.cgi?acc=LU00000000000000?amount=1000
Cookie: AS21389:6812HSADI:3991238
```

[4] http://en.wikipedia.org/wiki/Cross-Site_Request_Forgery

## SMTP over TLS

There are 2 major ways to used TLS with SMTP – STARTTLS and TLS from the beginning. With STARTTLS you connect to the SMTP port using plain text and then request a TLS connection using the command "STARTTLS".

T. ZOLLER (G-SEC) as well as W.VENEMA (Postfix) have researched this protocol independently, the following represents a summary of what is currently known. Mr. Venema has published a PDF that summarizes his views[5]. The document from VENEMA is unclear as to the prerequisites for a successful attack.

Discussions with M. VENEMA resulted in the following information which is based on a theoretical attack in-line with protocol requirements; the attack however requires the SMTP server to use a TLS engine that reads the data as soon as it arrives. VENEMA indicated STUNNEL would be a good candidate.

The following information is believed to be true during the time of writing:

## Protcol vulnerability matrix

Hypothesis: The attacker does **NOT have an account** on the SMTP server

| | |
|---|---|
| Attack theoretically possible if | TLS private cert authentication without SASL |
| | SMTP over TLS without SASL |

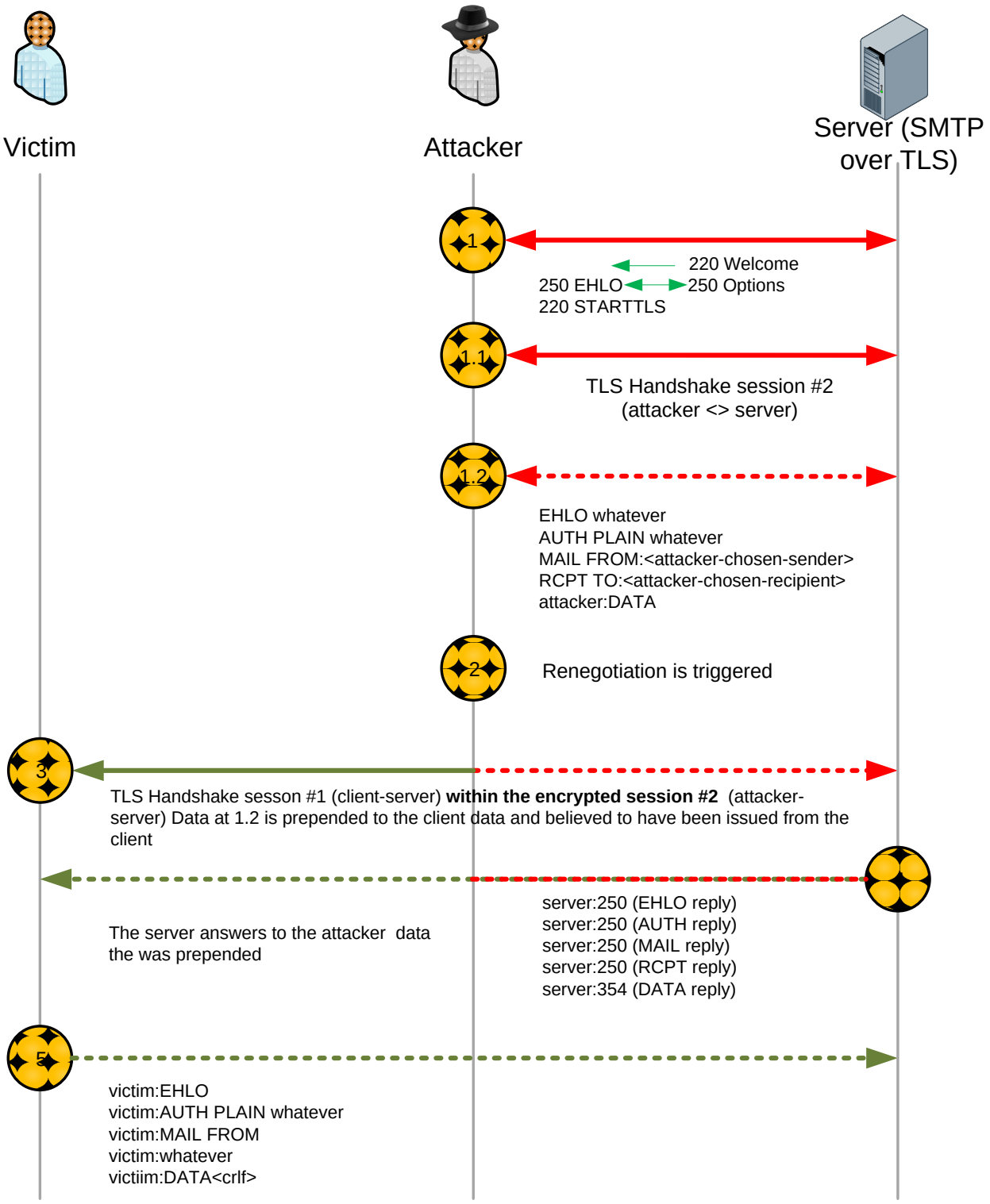
Hypothesis: The Attacker **has an account** on the SMTP server

| | |
|---|---|
| Attack theoretically possible if | TLS private cert authentication without SASL |
| | TLS private cert authentication with SASL |
| | SMTP over TLS with SASL |
| | SMTP over TLS without SASL |

Example of SASL : Auth plain, Auth MD5-CRAM

[5] http://www.porcupine.org/postfix-mirror/smtp-renegotiate.pdf

## SMTP STARTTLS (110)



## Details

This is a complex example of how this vulnerability could be used to exploit SMTP over TLS (STARTSSL) **if the attacker has an account**

1 Attacker connects to the SMTP server and initiates a TLS session (STARTTLS)

1.1 The attacker negotiates a new session performs a full TLS exchange

1.2 The attacker sends SMTP commands to the server but does not end the SMTP session, in this example the attacker controls the source and destination e-mail addresses.

2 Renegotiation is triggered

3 Attacker initiates a TLS session (TLS HELLO) and the victim performs a new TLS Handshake **over the previously established encrypted TLS session #2** (Attacker<>Server)

4 The TLS endpoint, due to the renegotiation has to take into the account the previously sent data (per spec), the endpoint believes the previous data (1.2) to have been send from the same client

As such the client now receives the answers from the attacker injected commands (note this is a way to detect this attack on the client-side).

5 The victim SMTP client now issues his commands to send mail – Those commands end up in the BODY of the mail previously started by the attacker.

The SMTP server receives:

```
EHLO whatever
AUTH PLAIN whatever
MAIL FROM:<attacker-chosen-sender>
RCPT TO:<attacker-chosen-recipient>
attacker:DATA
victim:EHLO
victim:AUTH PLAIN whatever
victim:MAIL FROM
victim:whatever
victiim:DATA<crlf>
```

As such the :<attacker-chosen-recipient> receives a mail containing the authentication data aswell as the other data.

## Client side attack detection

Contrary to HTTPS protocol the client has a way to detect that he was attacked at the application layer as the server replys arrive before the victim even sent the commands

## Important Note

To our knowledge POSTFIX is not affected by this vulnerability.

## The Impact on protocols using TLS

The impact of this vulnerability is different from one protocol to another. Several stateless protocols like HTTP for instance, merge both sessions into one, making it possible for the attacker to inject arbitrary plain text into the stream that is processed by the end stream as coming from the same destination

This breaks a principal assumption made by application developers and has impacts on innumerable number of custom implementations.

## Summary

| Protocol | Impact analysis available | Current status |
|---|---|---|
| HTTPS | Yes | Vulnerable to a certain degree, impact depends on application level logic and structure of the HTTP requests. |
| EAP-TLS | Online discussions | Believed to not be vulnerable |
| IMAPS | No | Unknown |
| POP3S | No | Unknown |
| LDAPS | No | Unknown |
| SMTP | Yes | Vulnerable only if certain requirements are met |

| Application | Impact analysis available | Current status |
|---|---|---|
| OpenVPN | Partially (vendor) | Not vulnerable, does not rely on openssl session capabilities – session handling was hardened after disclosure reports[6] |
| Tomcat | Partially (vendor) | Vulnerable[7] - mitigations exist |
| Apache | Available | Vulnerable – short term patch available[8] |
| IIS 7 <=7.5 | Available | Vulnerable - |
| GNUtls | Available | Vulnerable – patch status unknown, IETF proposal currently being implemented |
| OpenSSL | Available | Vulnerable – short term patches available, proposal currently being implemented |
| JSSE / NSS | No | May be vulnerable[9] |
| Citrix Secure Gateway 3.1 | No | Vulnerable |

Please refer to VU#120541 for an updated list of applications

[6] http://www.pubbs.net/openvpn/200911/19535/
[7] http://www.mail-archive.com/users@tomcat.apache.org/msg69335.html
[8] http://marc.info/?l=apache-httpd-announce&m=125755783724966&w=2
[9] http://blogs.sun.com/security/entry/vulnerability_in_tls_protocol_during

### EAP-TLS

EAP-TLS is not believed to be vulnerable if implemented as per specification[10].

- There is no application layer protocol involved when EAP-TLS is executed
- Only the TLS key material is used, the tunnel is not used.
- EAP re-authentication not the same as TLS renegotiation which is executed in the previous TLS tunnel

## Proposed IETF solution

The IETF draft proposed by E. Rescorla, M. Ray, S. Dispensa, N. Oskov offers an elegant way to solve the problem.

The Draft proposes a new TLS extension that cryptographically binds TLS sessions to clients and further allows informing clients about renegotiations. Furthermore the proposed solution allows working with a defined rule set that allows either - Never to renegotiate - Only renegotiate if TLS negotiation extension is being used or Renegotiate anyways

As to our information all major vendors are currently implementing above proposed solution.

## Vulnerability requirements

The preconditions for a TLS or SSLv3 connection to be vulnerable are

1. The server acknowledges and accepts full TLS renegotiations in the middle of a connection and after the initial handshake
   **and**
2. The server assumes that both TLS sessions were negotiated with the same client
   **and**
3. The server treats both sessions as one and merges them at the application layer

As such this vulnerability might not been seen as a vulnerability in TLS but the as the bad choice to merge two different requests together by the endpoint.

[10] http://www.ietf.org/mail-archive/web/tls/current/msg04109.html

## Patching TLS

From the conditions that emerged in "Vulnerability conditions" the patching requirements might be:

### Client

- Mid-term : Implement the IETF proposal for a TLS extension tracking and handling renegotiation requests[11] (draft-rescorla-tls-renegotiation-00.txt)

### Server

- Short-term : Remove renegotiation capabilities altogether
- Mid-term : Implement the IETF proposal for a TLS extension tracking and handling renegotiation requests[12] (draft-rescorla-tls-renegotiation-00.txt)

## Patching SSLv3

The only way to fix the renegotiation vulnerability for SSLv3 is to disable renegotiation on the server side completely. SSLv3 **does not support extensions** and as such cannot use the proposed extension mentioned above.

## Testing for a renegotiation vulnerability

The toolset provided by Openssl[13] offers the simplest way to test whether a server allows for client-side renegotiation in the established tunnel. Note: This doesn't necessarily mean that the application beneath is vulnerable to attacks over this channel, but indicates the server allows attacks to happen.

### Generic Example

```
Openssl s_client –connect yourserver.com:443
R (Triggers renegotiation – if this works, the server accepts renegotiations
within an existing TLS session Requirement 1)
GET /clientcontrolled.html http\1.0
```

[11] https://svn.resiprocate.org/rep/ietf-drafts/ekr/draft-rescorla-tls-renegotiate.txt
[12] https://svn.resiprocate.org/rep/ietf-drafts/ekr/draft-rescorla-tls-renegotiate.txt
[13] http://www.openssl.org/

## Conclusions

The vulnerability lies within the core of TLS and SSLv3, and will rear its ugly head for years to come; the custom applications that are potentially vulnerable are innumerable.

### Servers

- Servers that do allow mid-connection renegotiations are vulnerable
- Applications that handle 2 TLS sessions as coming from the same client are vulnerable

### Clients

- Clients have no means (pre TLS extension) to check if a renegotiation is happening and are vulnerable

## Sources


1. http://www.securityfocus.com/bid/36935
2. https://svn.resiprocate.org/rep/ietf-drafts/ekr/draft-rescorla-tls-renegotiate.txt
3. https://bugzilla.mozilla.org/show_bug.cgi?id=526689
4. http://blog.ivanristic.com/2009/11/ssl-and-tls-authentication-gap-vulnerability-discovered.html
5. http://www.leviathansecurity.com/pdf/ssltlstest.zip
6. http://extendedsubset.com/renegotiating_tls_20091104_pub.zip
7. https://bugzilla.redhat.com/show_bug.cgi?id=533125
8. http://www.mail-archive.com/users@tomcat.apache.org/msg69335.html
9. http://www.apache.org/dist/httpd/patches/apply_to_2.2.14/CVE-2009-3555-2.2.patch
10. http://sid.rstack.org/blog/index.php/373-tls-tout-le-monde-en-parle-pourquoi-pas-moi
11. https://www.mikestoolbox.net/
12. http://extendedsubset.com/
13. http://extendedsubset.com/Renegotiating_TLS.pdf
14. http://extendedsubset.com/Renegotiating_TLS_pd.pdf
15. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3555


## Thanks

We would like to thank Wietse Venema, Alexandre Dulaunoy, Noam Rathaus, j.clausing and Simon Zuckerbraun

## Disclaimer

Information is believed to be accurate by the time of writing. As this vulnerability is complex this document may be prone to revisions in the future.