TLS renegotiation authentication GAP

Yes, it is really a serious vulnerability

Agenda

- Why do we use TLS/SSL?
  - A typical SSL session
- What is renegotiation?
  - An SSL renegotiated session
- The attack
- The consequences
  - Ordering pizza
  - Stealing twitter passwords
- The (lack of) patches

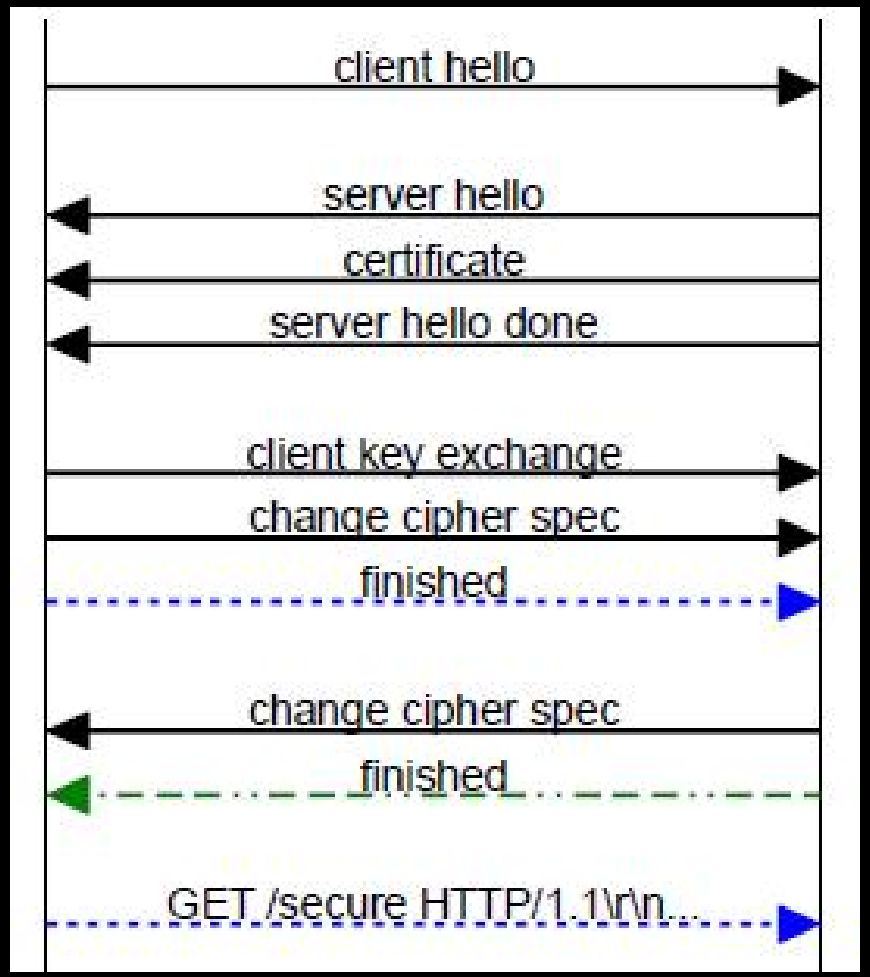
Why do we use TLS/SSL?

» TLS is intended to provide:
  - Confidentiality
  - Integrity
  - Non repudiation

» Intended to secure communication over an entrusted channel

» Certificates prove identity of Server (and client) and should make MitM attacks impossible

A typical SSL Session



What is renegotiation?

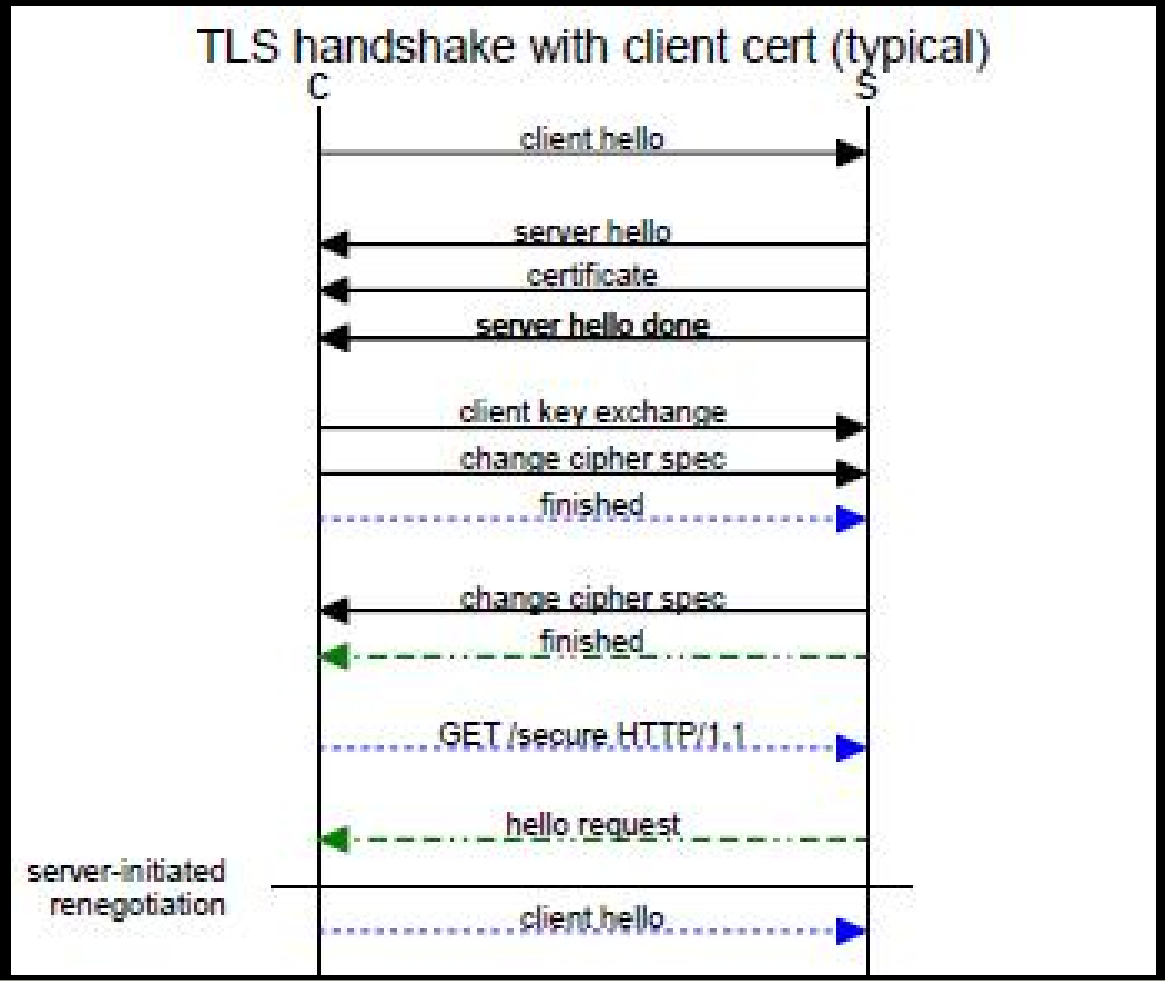
Client and server are allowed to initiate renegotiation of the session encryption in order to:

- Refresh keys
- Increase authentication
- Increase cipher strength
- Any other reason they see fit

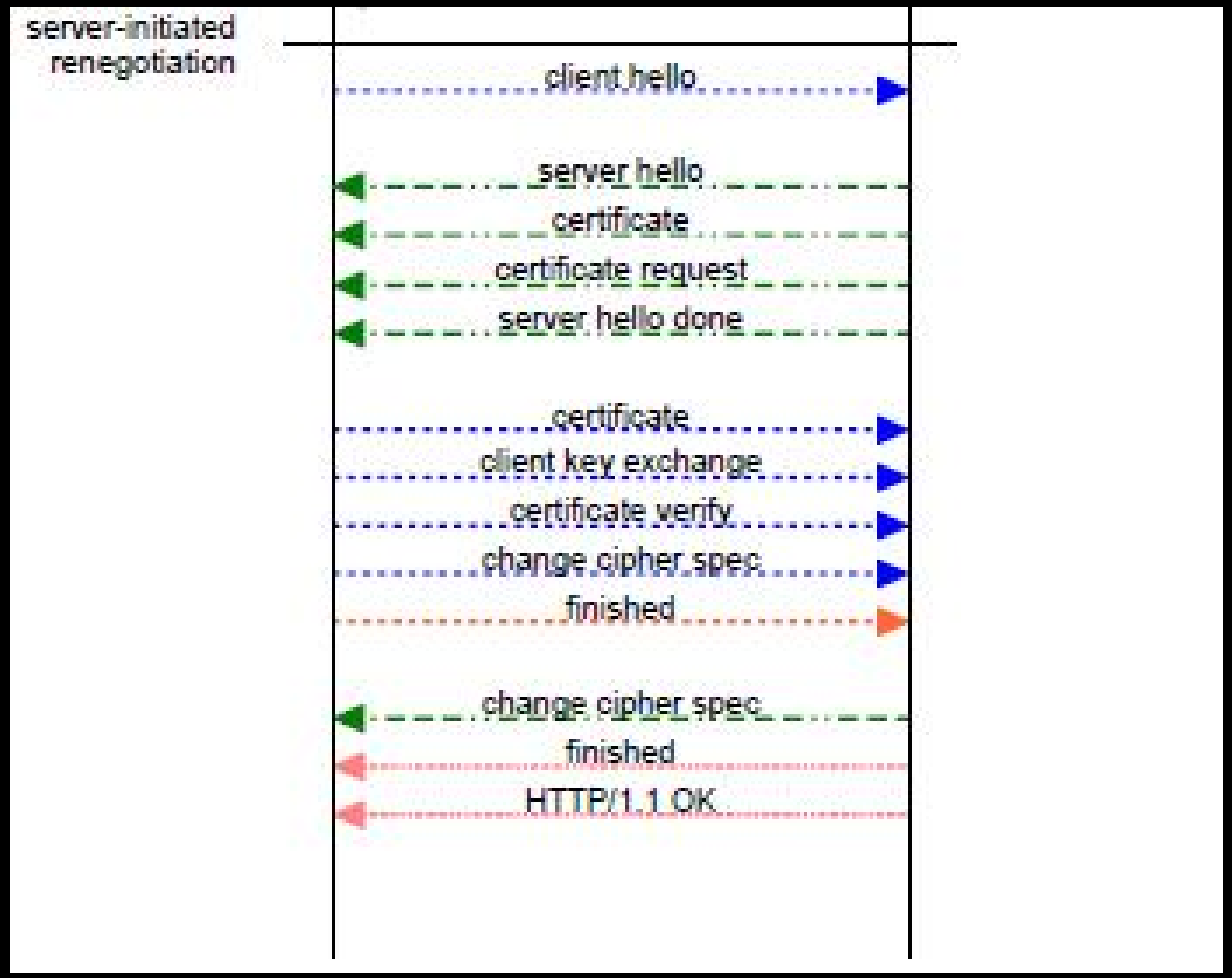
Renegotiation happens if client or server sends a hello message

E.g. a webserver has some directories that require certificate authentication and some that don’t

A typical renegotiated SSL session

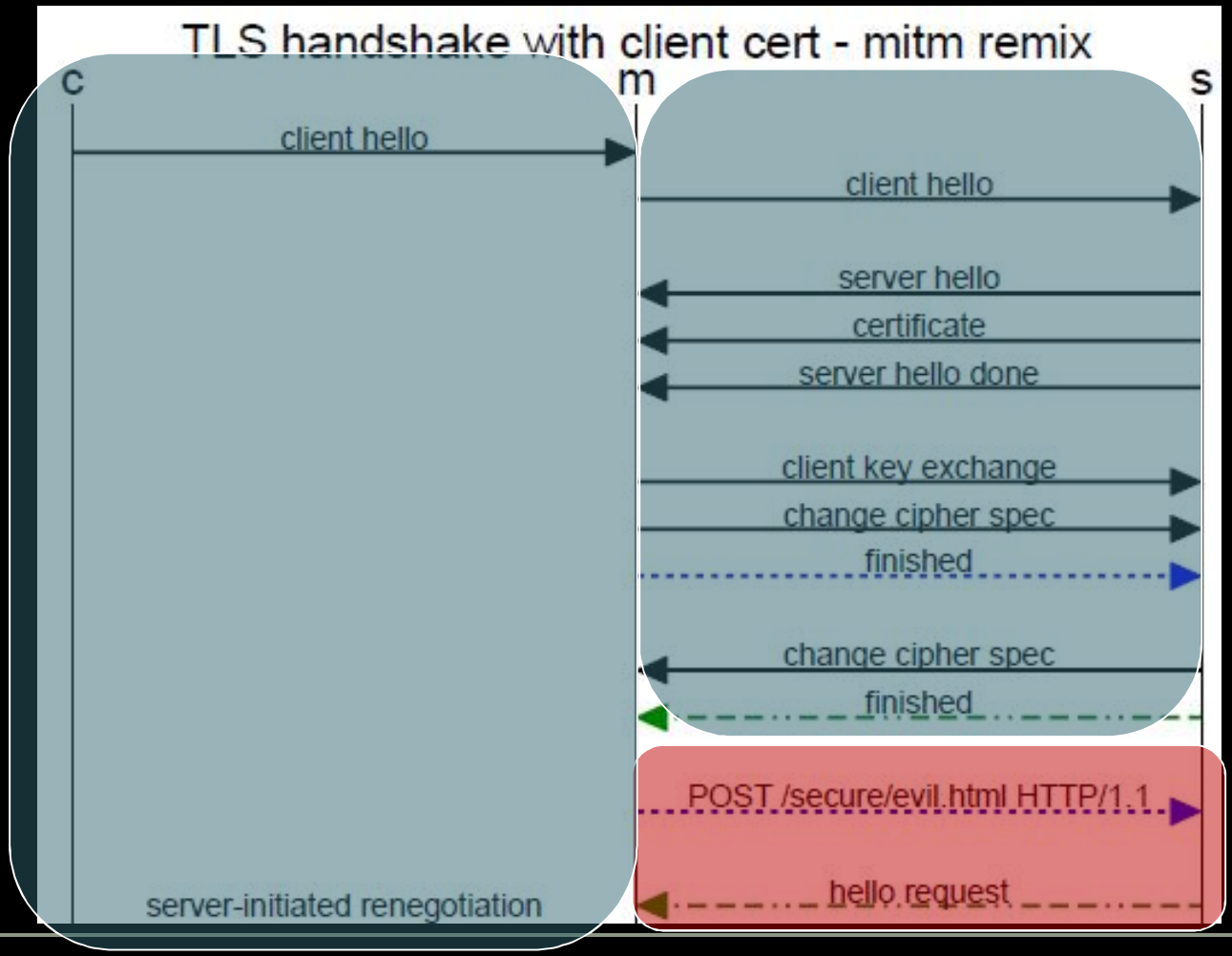


A typical renegotiated SSL session



The attack

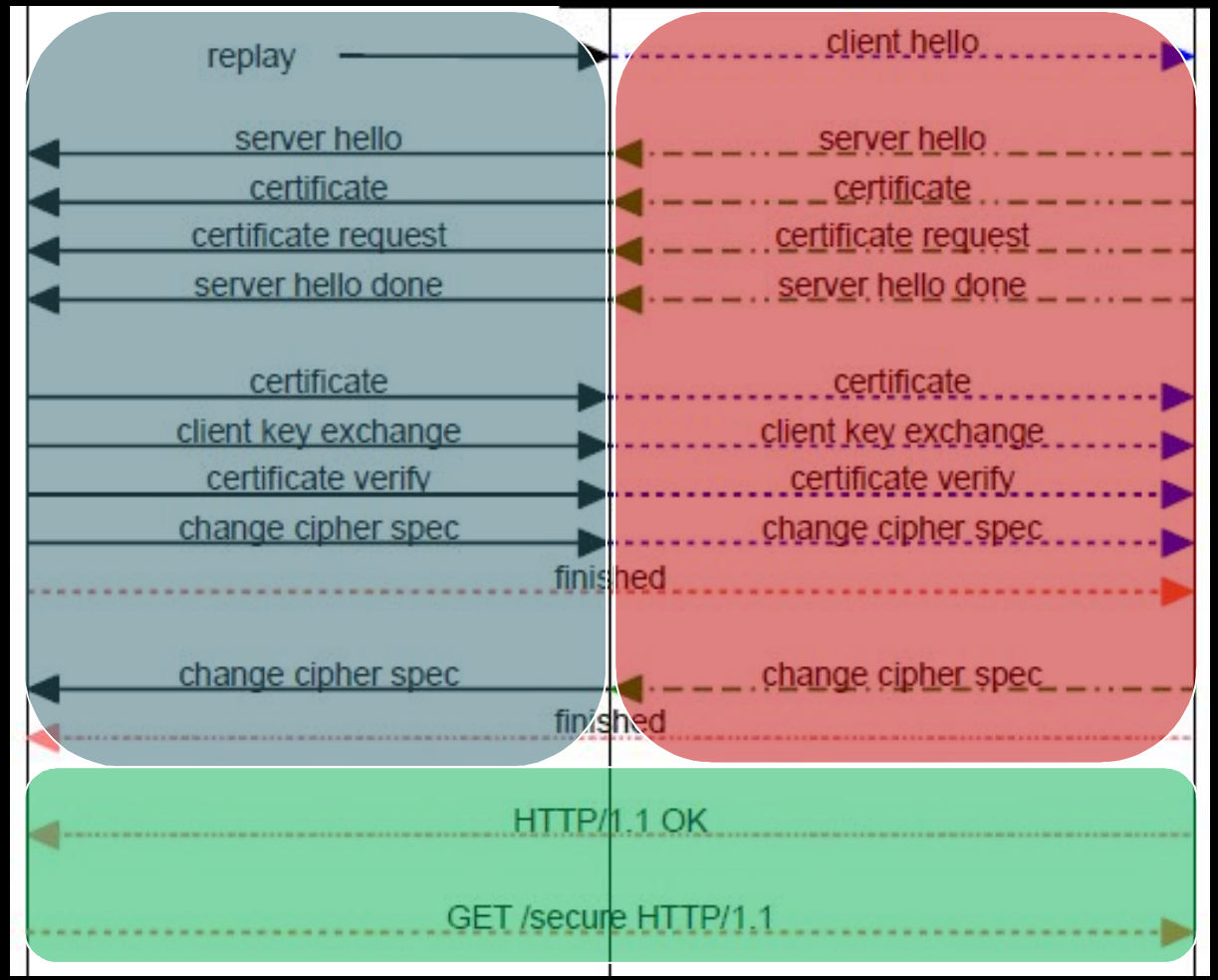
No cipher context



MitM cipher context

The attack

No cipher context



MitM cipher context

Client cipher context

The consequences

Integrity of the original request

» Attacker can insert data (text) in front of anydata sent by client

Integrity of reply

» Attacker cannot modify reply, but can prevent that reply reaches client

Confidentiality

» Attacker cannot decrypt request or reply

Non repudiation

» It can not longer be guaranteed that therequest is the request intended by client, even when client certificates are used

The consequences - Ordering pizza

Normal pizza ordering query:
GET /pizza?toppings=ansjovis;address=andrzejs%20place\n
Cookie: andrzejwillpayyourpizza\n
\n

Attacker sends:
GET /pizza?toppings=pepperoni;address=franks%20place\n
X-ignore-next:

Query becomes:
GET /pizza?toppings=pepperoni;address=franks%20place\n
X-ignore-next: GET /pizza?toppings=ansjovis;address=andrzejs%20place\n
Cookie: andrzejwillpayyourpizza\n
\n

The consequences - Stealing twitter passwords

Confidentiality wasn't broken, but still passwords where stolen. Here's how I think they did it…

Victim message:
POST /statusses/update.xml\n
Authorisation: Basic <victims base 64 creds>\n
User-agent: <some user agent>
Status=Victims status\n
Etc…..

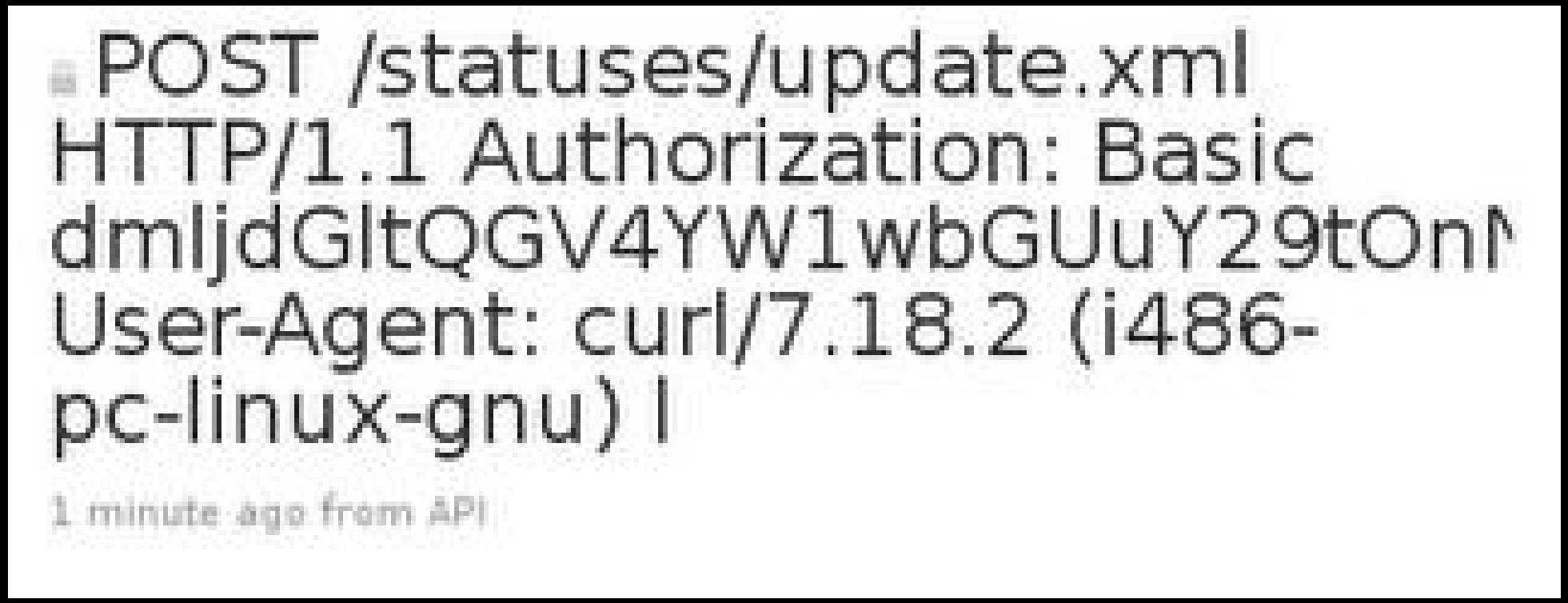
Injected message:
POST /statusses/update.xml\n
Authentication-basic: <attacker creds>\n
Status=

Consequences - Stealing twitter passwords

Becomes:

POST /statusses/update.xml\n
Authentication-basic: <attacker creds>\n
Status=POST /statusses/update.xml\n
Authorisation: Basic <victims base 64 creds>\n
User-agent: <some user agent>
Status=Victims status\n
Etc…..

The consequences - Stealing twitter passwords



The (lack of) patches (as of 18-11-2009)

IETF
- There is a proposed draft that addresses this issue (Draf Rescorla TLS renegotiation 01)

OpenSSL
- Workaround (disable renegotiation) is available for download (OpenSSL 0.9.8l)
- Fix (based on rescorla draft) is being tested (OpenSSL 0.9.8m)

Microsoft
- Currently testing interoperability
- ISS6 and IIS7 not vulnerable for client initiated renegotiation

Cisco
- Initial testing stage

F5
- Workaround is availble (disable renegotiation completly)
- Fix in initial testing stage

Mozilla/Firefox/Thunderbird/NSS
- Initial testing

Sun
- Initial testing

GNU utils
- Mostly not affected
- Initial testing

RSA BSAFE suite
- Limited beta

Opera
- Initial testing

Sources

March Ray:

- http://extendedsubset.com/
- http://extendedsubset.com/Renegotiating_TLS.pdf
- http://extendedsubset.com/Renegotiating_TLS_pd.pdf

Phonefactor

- http://www.phonefactor.com/sslgap/ssl-tls-authentication-patches

Ekron

- http://www.educatedguesswork.org/2009/11/understanding_the_tls_renegoti.html

Anil Kurmus

- http://www.securegoose.org/2009/11/tls-renegotiation-vulnerability-cve.html

F5

- http://devcentral.f5.com/weblogs/cwalker/archive/2009/11/06/20-lines-or-less-31-ndash-traffic-shaping-header-re-writing.aspx

Questions?

Want to know more?

Frank Breedijk

- Security Engineer at Schuberg Philis
- Author of Seccubus
- Blogger for CupFigther.net

Email: fbreedijk@schubergphilis.com
Twitter: @seccubus
Blog: http://cupfighter.net
Project: http://seccubus.com