Modelling ciphersuite and version negotiation in the TLS Protocol

Queensland University of Technology
Brisbane Australia

BENJAMIN DOWLING AND DOUGLAS STEBILA

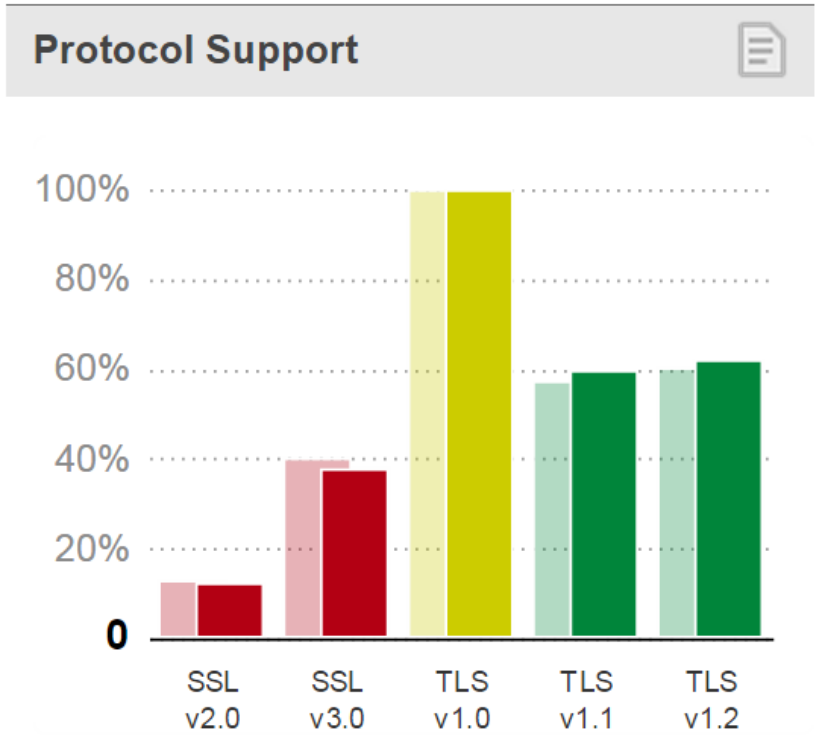
Outline

1. Motivation
2. Negotiation in the TLS Protocol
3. Modelling negotiation in a provable security framework
4. Analysis of TLS ciphersuite and version negotiation
5. Conclusions

Motivation

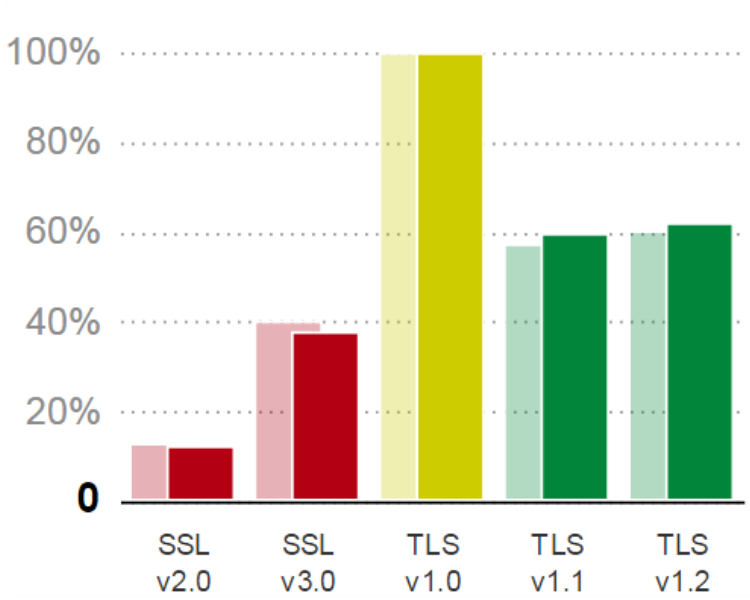
- TLS implementations have complex functionality
- Algorithmic agility is desired to increase interoperability
- Interoperability can affect security

Motivation

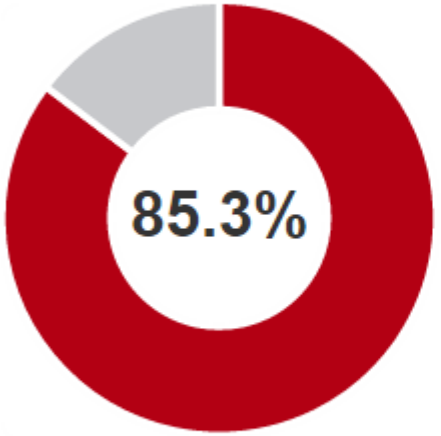

Protocol Support
100%
80%
60%
40%
20%
0
SSL v2.0
SSL v3.0
TLS v1.0
TLS v1.1
TLS v1.2

Motivation

Protocol Support



BEAST Attack



Sites that are vulnerable to the BEAST attack

125,153

+ 1.0 %

Version Downgrade Attacks

- Moeller and Langley - TLS extension to prevent downgrades
- SSLv2 provides **very** limited authentication properties
- SSLv3 vulnerable to BEAST and POODLE attacks

Version Downgrade Attack

TLS 1.1

- Client attempts handshake
- Version Failure Response (unauthenticated)

TLS 1.0

- Client attempts handshake
- Version Failure Response (unauthenticated)

SSLv3

- Client attempts handshake
- Success! (but not really…)

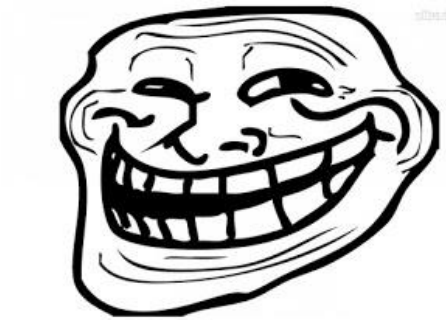
Ciphersuite Downgrade Attacks

- FREAK attack – (Beurdouche, Bhargavan, Delignat-Lavaud, Fournet, Kohlweiss, Pironti, Strub, Zinzindohoue, Zanella-Béguelin; 2015)
  - Implementation errors allow the negotiation of Export-RSA despite no indicated support

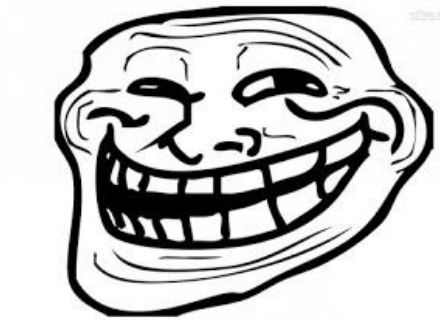
- Logjam Attack – (Adrian, Bhargavan, Durumeric, Gaudry, Green, Halderman, Heninger, Springall, Thomé, Valenta, VanderSloot, Wustrow, Zanella-Beguelin, and Zimmermann; 2015)
  - Protocol logic misinterprets export-DHE shares as “normal” DHE shares

Ciphersuite Downgrade Attacks

ClientHello:
bad_ciphersuite,
good_ciphersuite



Ciphersuite Downgrade Attacks



ClientHello’:
bad_ciphersuite

Ciphersuite Downgrade Attacks

ServerHello:
bad_ciphersuite

Ciphersuite Downgrade Attacks

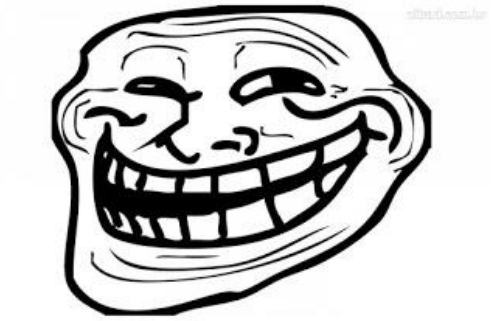
Finished: good ciphersuite, bad_ciphersuite,

Ciphersuite Downgrade Attacks

Finished’:
bad_ciphersuite

Ciphersuite Downgrade Attacks

Finished:
bad_ciphersuite



Observations

- Clearly negotiation from a family of protocols can affect security of the protocol as a whole

- What can we say about the security of the collection of protocols?

Talking about negotiation

- Treat the handshake as two phases:
  - A negotiation phase: common to all handshake runs
  - A sub-protocol phase: uses negotiated values to execute key-exchange/authentication, etc.

- "Optimal" negotiation:
  - Both parties have ordered list of preferences
  - Also have an "optimality function"
  - Negotiation is optimal if they output same value and it's the output of *opt(list, list')*

Using previous results

- Can see from downgrade attacks that security of the negotiation relates to the authentication of transcript

- Negotiation-Authentication Theorem:
  - Condition 1: All Negotiation Phase messages are in the session identifier
  - Condition 2: If no modification of messages, negotiation always “optimal”

Ciphersuite Negotiation “secure”

- 1. All negotiation messages contained in transcript
- 2. Ciphersuite negotiation optimal without active adversary

- If all ciphersuites result in secure authentication properties then negotiating to any given ciphersuite is secure

Version Negotiation (no fallback) “secure”

- 1. All negotiation messages contained in transcript
- 2. Version negotiation optimal without active adversary

- If all versions result in secure authentication properties then negotiating to any given version is secure

Version Negotiation (w/ fallback) “secure”

- 1. All negotiation messages contained in transcript?

Version Negotiation (w/ fallback) insecure

- 1. All negotiation messages contained in transcript?

NOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

Version Negotiation (w/ fallback) insecure

- 1. All negotiation messages contained in transcript?

NOOOOOOOOOOOOOOOOOOOOOOOOOO

- Negotiation occurs across multiple handshakes, session identifier is only the transcript of the most recent handshake

Version Negotiation (w/ SCSV) “secure”

- 1. All negotiation messages contained in transcript
- 2. Version negotiation optimal without active adversary

- HOWEVER: Non-contiguous support of TLS version (i.e. supporting 1.2 and 1.0 but not 1.1) can break version negotiation with SCSV

Conclusions

- When considering negotiation security, think:

Weakest Link Security

- Additionally:

Always authenticate everything

Thanks!

Questions?