On the Effective Prevention of TLS Man-In-The-Middle Attacks in Web Applications

Nikos Karapanos and Srdjan Čapkun, ETH Zurich

USENIX Security 2014
Server authentication is problematic
Server authentication is problematic

- Compromised CAs
Server authentication is problematic

- Compromised CAs
- Compromised server keys
Server authentication is problematic

- Compromised CAs
- Compromised server keys
- Users click through warnings
TLS Man-In-The-Middle (MITM)

Goal: Compromise user account

*Some of the icons used in this presentation were taken and adapted from opensecurityarchitecture.org*
TLS Man-In-The-Middle (MITM)

Goal: Compromise user account

*Some of the icons used in this presentation were taken and adapted from opensecurityarchitecture.org*
Goal: Compromise user account

*Some of the icons used in this presentation were taken and adapted from opensecurityarchitecture.org
TLS Man-In-The-Middle (MITM)

Goal: Compromise user account

TLS Channel IDs (Balfanz et al., IETF Internet Draft) proposed as a solution

*Some of the icons used in this presentation were taken and adapted from opensecurityarchitecture.org
1. We show an attack against TLS Channel IDs
   - extends usually considered attacker models
   - implemented and tested
1. We show an attack against TLS Channel IDs
   - extends usually considered attacker models
   - implemented and tested

2. We propose a new solution: SISCA (Server Invariance with Strong Client Authentication)
   - prevents MITM attacks even under server impersonation
   - prototype implemented
Solutions focus on either endpoint
Preventing TLS MITM: Overview

Solutions focus on either endpoint

Prevent server impersonation

Pinning, multipath probing
Solutions focus on either endpoint

Prevent user impersonation

TLS Channel ID-based authentication
Channel ID = public key of a private/public key pair
Channel ID = public key of a private/public key pair

TLS Channel ID-Based Client Authentication

Here is my channel ID, signed with the corresponding private key

www.example.com
Channel ID = public key of a private/public key pair
Channel ID = public key of a private/public key pair
Channel ID = public key of a private/public key pair

Here is my channel ID, signed with the corresponding private key

: TLS Channel IDs

www.example.com
Channel ID = public key of a private/public key pair

A Channel ID identifies the same “TLS channel” across different TLS connections

TLS Channel ID-Based Client Authentication

www.example.com
Channel ID = public key of a private/public key pair

A Channel ID identifies the same “TLS channel” across different TLS connections

Here is my channel ID, signed with the corresponding private key

: TLS Channel IDs
Channel ID = public key of a private/public key pair

A Channel ID identifies the same “TLS channel” across different TLS connections

: TLS Channel IDs
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login

Cookie presented over wrong channel!

TLS Channel IDs
Initial login (first login from a browser)

PhoneAuth (Czeskis et al., CCS 2012), FIDO Alliance U2F draft spec.

After initial login

Cookie presented over wrong channel!
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)

www.example.com

www.example.com
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)

www.example.com

www.example.com
MITM-Script-In-The-Browser (MITM-SITB)
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)

Executed within the target server's web origin
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)

Executed within the target server’s web origin
Our Attack

MITM-Script-In-The-Browser (MITM-SITB)

Executed within the target server’s web origin

www.example.com

www.example.com
MITM-SITB was missed by a number of proposals
MITM-SITB was missed by a number of proposals

- TLS Channel IDs (PhoneAuth, FIDO U2F)
- TLS client auth., SSL/TLS session-aware user auth. (Oppliger et al, Computer Communications 2006)
MITM-SITB was missed by a number of proposals

- TLS Channel IDs (PhoneAuth, FIDO U2F)

- TLS client auth., SSL/TLS session-aware user auth. (Oppliger et al, Computer Communications 2006)

- These solutions focus on client authentication but ignore server authentication.
  - Attacker impersonates the server and injects malicious but “trusted” client-side code
MITM-SITB was missed by a number of proposals

- TLS Channel IDs (PhoneAuth, FIDO U2F)
- TLS client auth., SSL/TLS session-aware user auth. (Oppliger et al, Computer Communications 2006)
- These solutions focus on client authentication but ignore server authentication.
  - Attacker impersonates the server and injects malicious but “trusted” client-side code

=> we cannot ignore server authentication
But…
Do we really need server authentication?

Insight
Do we really need server authentication?
Do we really need server authentication?

Insight

Conventional MITM prevented by Channel ID-based client auth.
Do we really need server authentication?

Insight

Conventional MITM prevented by Channel ID-based client auth.
Do we really need server authentication?

Insight

Conventional MITM prevented by Channel ID-based client auth.

MITM-SITB needs the browser to connect to two different entities
Do we really need server authentication?

Insight

1. Attacker server (inject code)

Conventional MITM prevented by Channel ID-based client auth.

MITM-SITB needs the browser to connect to two *different* entities
Do we really need server authentication?

**Insight**

1. Attacker server (inject code)

**Conventional MITM** prevented by Channel ID-based client auth.

**MITM-SITB** needs the browser to connect to two **different** entities

2. Legitimate server (access user account)
Do we really need server authentication?

Insight

Conventional MITM prevented by Channel ID-based client auth.

OUR PROPOSAL:
ensure that the browser does not connect to different entities!
This Invariance is Enforced by the Browser
This Invariance is Enforced by the Browser
This Invariance is Enforced by the Browser
This Invariance is Enforced by the Browser

1st TLS

TLS

TLS

www.example.com

www.example.com

www.example.com

✅
This Invariance is Enforced by the Browser
This Invariance is Enforced by the Browser
This Invariance is Enforced by the Browser
Example of Realizing Server Invariance

1. Initialization (first connection)
2. Invariance verification

www.example.com
Example of Realizing Server Invariance

1. Initialization (first connection)
2. Invariance verification

$\text{store: } [r_b, r_s]$

---

www.example.com

$\text{store: } [, r_b, r_s]$
Example of Realizing Server Invariance

1. Initialization (first connection)
2. Invariance verification

- \( r_b \)
- store: \([r_b, r_s]\)
- \( r_s = ? r_s' \)
- lookup: \( r_s' \) from \([r_b, r_s]\)

\[
\text{www.example.com}
\]

\[
\begin{align*}
&\text{store: } [\text{certificate}, r_b, r_s] \\
&\text{lookup: } r_s' \text{ from } [\text{certificate}, r_b]
\end{align*}
\]
Important Insight

TLS MITM prevention
TLS MITM prevention

Weak client authentication
- passwords, conventional HTTP
- cookies, OTP, …

Server authentication
- certificate pinning, certificate transparency, …
TLS MITM prevention

- Weak client authentication
  - passwords, conventional HTTP cookies, OTP, …

- Server authentication
  - certificate pinning, certificate transparency, …

- Strong client authentication
  - Channel ID-based (FIDO U2F, channel-bound cookies), …

- Server invariance
Important Insight

TLS MITM prevention

Weak client authentication + Server authentication
- passwords, conventional HTTP cookies, OTP, …
- certificate pinning, certificate transparency, …

Strong client authentication + Server invariance
- Channel ID-based (FIDO U2F, channel-bound cookies), …
- SISCA
• In web, servers can ask clients to execute arbitrary code
  - needs to be taken into account in protocol and system analysis

• TLS Channel IDs vulnerable to MITM-SITB attacks
• In web, servers can ask clients to execute arbitrary code
  - needs to be taken into account in protocol and system analysis

• TLS Channel IDs vulnerable to MITM-SITB attacks

• To prevent MITM attacks we need either:
  - server authentication or…
  - server invariance with Channel ID-based client authentication

• Server invariance is easier to achieve than server authentication
  => we propose SISCA: Server Invariance with Strong Client Authentication
Thank you for your attention!
Any Questions?

knikos@inf.ethz.ch