You shall not password !

An extensive analysis of multi-factor authentication protocols

Charlie Jacomme, Steve Kremer

CSF 2018

Introduction

Our goal

Secure Authentication:

Every accepted login by the server and coming from some computer has been initiated on the very same computer by the user.

Common solution: login / password

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Everybody uses the same password everywhere !

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Everybody uses the same weak password everywhere !

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Everybody uses the same weak password everywhere !

"1234", "password", "qwerty"

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Everybody uses the same weak password everywhere !
"1234", "password", "qwerty"

Requirement to add special characters or on length does not work

Passwords are bad

Passwords are compromised:

- Database leaks
- Phishing
- Keyloggers

Everybody uses the same weak password everywhere !

"1234", "password", "qwerty"

Requirement to add special characters or on length does not work

"123456!", "p@ssword1", "Qwerty"

Second Factor authentication

The current solution

Use a second factor to confirm login, either a smartphone or a dedicated token.

Second Factor authentication

The current solution

Use a second factor to confirm login, either a smartphone or a dedicated token.

Protocols we studied:

- Google 2 Step (Verification code, One Tap, Double Tap)
- FIDO's U2F (Google, Facebook, Github, Dropbox,...)

Main ideas

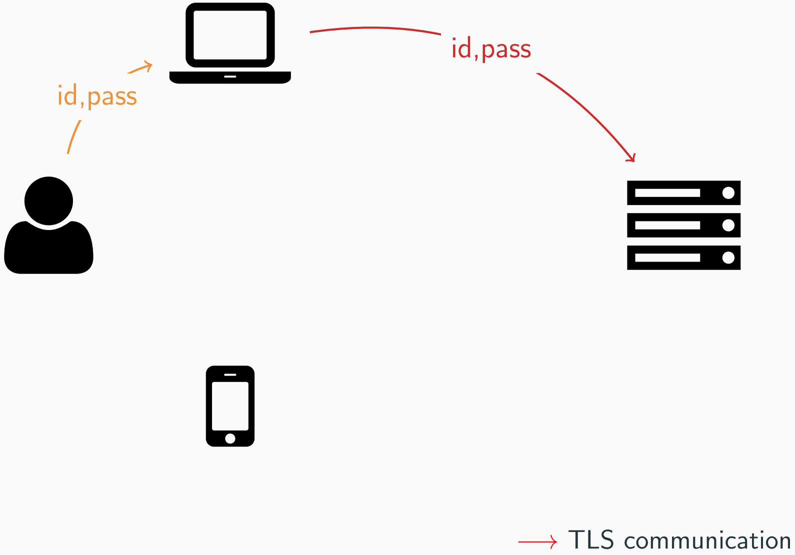
A case study of Google 2 Step and FIDO's U2F

- Many different threat models (malwares, phishing, human errors...)
- Automated analysis of all scenarios

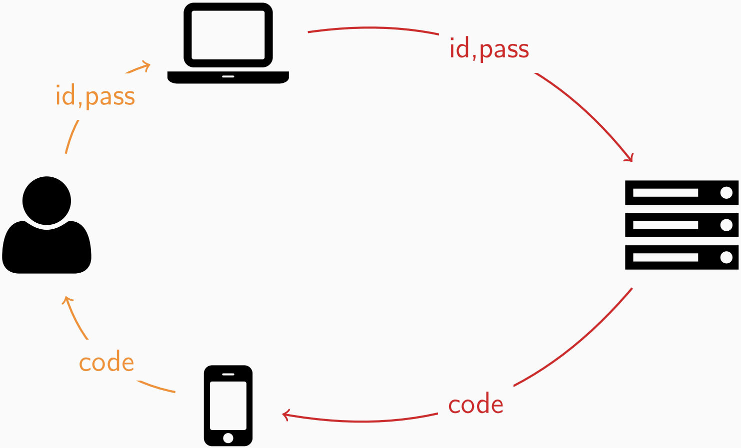
$\rightarrow$ 6 172 (non-redundant) scenarios analysed by PROVERIF in 8 minutes

Presentation of the protocols

Google 2 Step - Verification Code

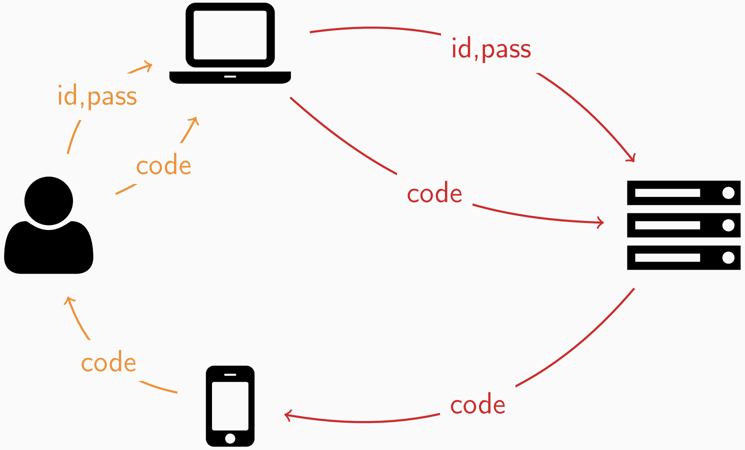


Google 2 Step - Verification Code

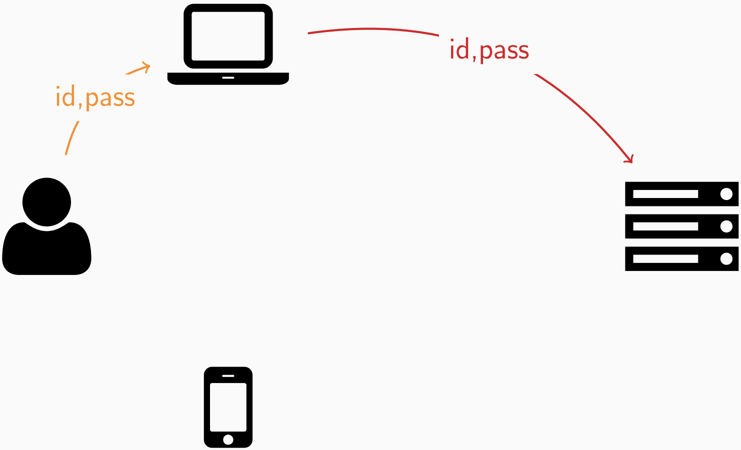


$\longrightarrow$ TLS communication

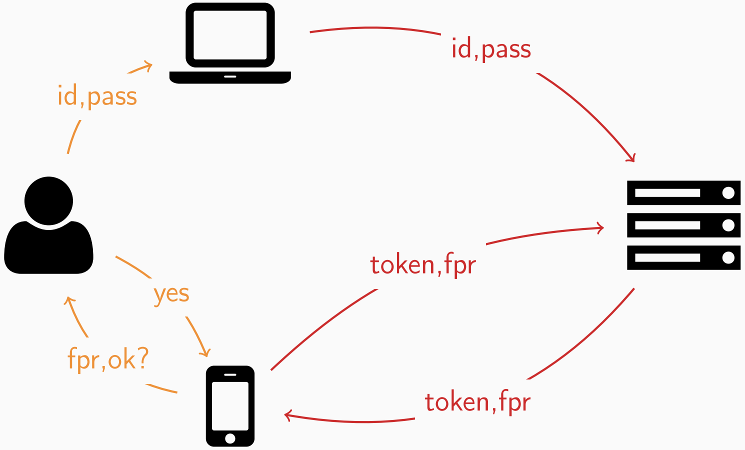
Google 2 Step - Verification Code



Google 2 Step - One Tap

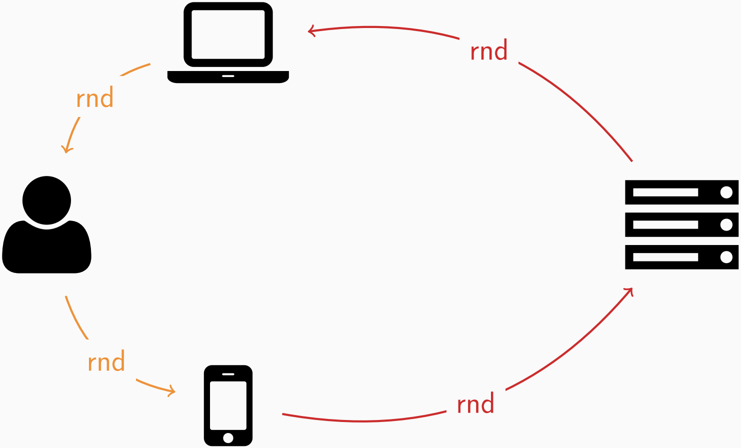


Google 2 Step - One Tap



fpr : IP,location, OS,...

Google 2 Step - Double Tap



FIDO

A token with cryptographic capabilities

- A public key is registered server side.
- On login, a challenge containing a random nonce, the origin and the TLS sid is signed.

I trust this computer

An option provided by major companies (Google, Facebook,...):

I trust this computer = disable second factor

I trust this computer

An option provided by major companies (Google, Facebook,...):

I trust this computer = disable second factor

It must be taken into account in the analysis

Threat model

Goals

First hypothesis

The user password has been compromised

Goals

First hypothesis

The user password has been compromised

Goal

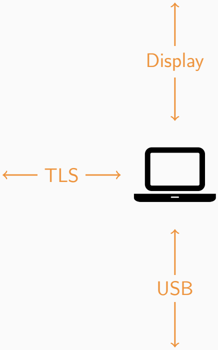
Consider many different scenarios :

- Malware on the computer
- Malware on the phone
- Human erros (Phishing, No Compare)
- Fingerprint Spoofing

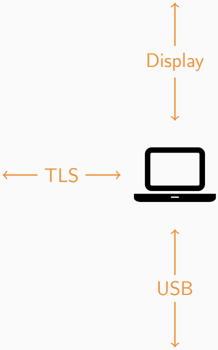
What guarantees from different protocols under different threats?

Modelling Malwares

Device = set of interfaces



Device = set of interfaces



Access levels

Read Only or Read Write

$$\begin{tikzcd}
 & \text{Out RW} \arrow[dl] \arrow[dr] & \\
\text{In RW} \arrow[dr] & & \text{Out RO} \arrow[dl] \\
 & \text{In RO} \arrow[d] & \\
 & \text{NA} &
\end{tikzcd}$$

Scenarios

Notations

- Malware : $\mathcal{M}^{\text{interf}}_{\text{in}:acc1,out:acc2}$
- Phising : PH
- Fingerprint Spoofing : FS
- No Compare : NC

Scenarios

Notations

- Malware : $\mathcal{M}^{\text{interf}}_{\text{in}:acc1,out:acc2}$
- Phising : PH
- Fingerprint Spoofing : FS
- No Compare : NC

Examples

- Keylogger : $\mathcal{M}^{\text{usb}}_{\text{in}:\mathcal{RO}}$
- Wifi Hotspot : FS PH
- Broken TLS encryption : $\mathcal{M}^{\text{tls}}_{\text{io}:\mathcal{RW}}$

Modeling in Proverif

TLS modeling

- A set of identities : id_{server}, $id_{user's\ computer}$, ...
- A private function symbol tls

TLS modeling

- A set of identities : id_{server}, $id_{user's\ computer}$, ...
- A private function symbol tls

TLS :=
Asynchronous communications over channel $tls(id_{client}, id_{server})$

TLS modeling

- A set of identities : id_{server}, $id_{user's\ computer}$, ...
- A private function symbol tls

TLS :=

Asynchronous communications over channel $tls(id_{client}, id_{server})$

If id_{client} or id_{server} is compromised, we give $tls(id_{client}, id_{server})$ to the attacker

Malwares

Read only access to some channel ch :

$$\mathbf{in}(ch, x).P \rightarrow \mathbf{in}(ch, x).\mathbf{out}(a, x).P$$

or

$$\mathbf{out}(ch, x).P \rightarrow \mathbf{out}(a, x).\mathbf{out}(ch, x).P$$

Malwares

Read only access to some channel ch :

$$\mathbf{in}(ch, x).P \to \mathbf{in}(ch, x).\mathbf{out}(a, x).P$$

or

$$\mathbf{out}(ch, x).P \to \mathbf{out}(a, x).\mathbf{out}(ch, x).P$$

Read write access to ch :

$$P \to \mathbf{out}(a, ch).P$$

Human errors

No compare

Remove some checks

Phishing

The server's url (id_{server}) is chosen by the attacker.
$\rightarrow$ The human may check or not that it is indeed the server he wishes to contact.

Fingerprint Spoofing

Fingerprint

A function symbol $fpr(id)$

$\rightarrow$ a server may obtain $fpr(id_{client})$ from $tls(id_{client}, id_{server})$

Fingerprint Spoofing

Fingerprint

A function symbol $fpr(id)$

$\rightarrow$ a server may obtain $fpr(id_{client})$ from $tls(id_{client}, id_{server})$

Spoofing

$$\mathtt{fpr}(\mathtt{spoof}_{\mathtt{fpr}}(\mathtt{fpr}(c))) = \mathtt{fpr}(c)$$

Analysis

Properties

Three types of login

- *untrusted login* login on an untrusted computer
- *trusted login* login on a trusted computer; sets "trust this computer" option
- *cookie login* login after "trust this computer" option enabled

Properties

Three types of login

- *untrusted login* login on an untrusted computer
- *trusted login* login on a trusted computer; sets "trust this computer" option
- *cookie login* login after "trust this computer" option enabled

Three properties

$$accept_x(id) \Longrightarrow_{\text{inj}} request_x(id) \qquad x \in \{u, t, c\}$$

Every accepted login was preceded by a distinct login request by the human.

One file = one protocol with all scenarios

```
let Device =
    in(d_in,(token));
      #if defined(D_I_RO) && !defined(D_I_RW)
      out(a,(token));
      #endif
    out(d_out,(token))
    |
        (
          ...
```

Methodology

A bash script

- takes a combination of attacker capabilities as input
- generates the proverif file

A python script

- runs proverif for all pertinent combinations of scenarios
- generate the result table

Analysis of Google 2 step protocols

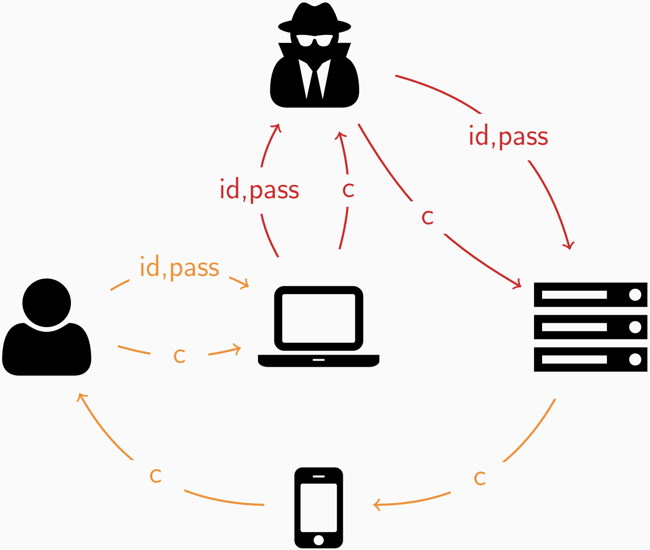
Threat Scenarios	g2V	g2OT	g2OT$^{\mathsf{fpr}}$
	✔	✖	✔
PH	✖	✖	✔
NC	✔	✖	✖
FS	✔	✖	✖
PH NC	✖	✖	✖
PH FS	✖	✖	✖
$\mathcal{M}^{\mathsf{dev}}_{\mathsf{in}:\mathcal{RO}}$	✖	✖	✔
$\mathcal{M}^{\mathsf{t-dis}}_{\mathsf{io}:\mathcal{RO}}$	✔	✖	✔
$\mathcal{M}^{\mathsf{t-tls}}_{\mathsf{io}:\mathcal{RO}}$	✖	✖	✔
$\mathcal{M}^{\mathsf{t-usb}}_{\mathsf{in}:\mathcal{RO}}$	✖	✖	✔
$\mathcal{M}^{\mathsf{dev}}_{\mathsf{in}:\mathcal{RW}}$	✖	✖	✖
$\mathcal{M}^{\mathsf{t-tls}}_{\mathsf{io}:\mathcal{RW}}$	✖	✖	✗✓✗
$\mathcal{M}^{\mathsf{t-usb}}_{\mathsf{in}:\mathcal{RW}}$	✖	✖	✓✓✗

...

Analysis of Google 2 step - Verification code

- It is secure if the attacker only knows the password
- in any other cases...

Attack under Keylogger or Phishing or Malware



Analysis of Google 2 step - One Tap

- Without fingerprint, never secure : one can easily validate an attacker session

Adding the display

Recommendation:

Display (via SMS or on the smartphone screen) additional info:

- fingerprint (IP,locations, computer model).
- the type of login desired.

Benefits:

- avoids attacks changing the login type (e.g. replacing an untrusted, by trusted login)
- avoids attacks where attacker is able to spoof a fingerprint

Adding the display

Threat Scenarios				$g2V^{fpr}$	$g2V^{dis}$	$g2OT^{dis}$	$g2DT^{dis}$
PH				✔	✔	✔	✔
PH	FS			✖	✗✓✓	✖	✗✓✓
PH	FS	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$		✖	✖	✖	✗✓✗
PH	FS	$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RO}}$		✖	✖	✖	✗✓✓
PH	FS	$\mathcal{M}^{\mathbf{t-dis}}_{\mathbf{io}:\mathcal{RW}}$		✖	✗✓✓	✖	✖
		$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$		✔	✔	✓✓✓	✔
		$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RO}}$		✔	✔	✓✓✓	✔
		$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RW}}$		✗✓✗	✓✓✗	✓✓✗	✓✓✗
		$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RW}}$		✓✓✗	✓✓✗	✓✓✗	✓✓✗
		$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RW}}$	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$	✓✓✗	✓✓✗	✓✓✗	✓✓✗
	FS	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$		✖	✓✗✗	✖	✓✓✗
	FS	$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RO}}$		✖	✓✗✗	✖	✔
	FS	$\mathcal{M}^{\mathbf{t-dis}}_{\mathbf{io}:\mathcal{RW}}$		✔	✔	✖	✓✗✗
	FS	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RW}}$		✖	✓✗✗	✖	✓✗✗
	FS	$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RW}}$		✖	✓✗✗	✖	✓✗✗
	FS	$\mathcal{M}^{\mathbf{t-dis}}_{\mathbf{io}:\mathcal{RW}}$	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$	✖	✓✗✗	✖	✓✗✗
	FS	$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RO}}$	$\mathcal{M}^{\mathbf{t-dis}}_{\mathbf{io}:\mathcal{RW}}$	✖	✓✗✗	✖	✓✗✗
	FS	$\mathcal{M}^{\mathbf{t-usb}}_{\mathbf{in}:\mathcal{RW}}$	$\mathcal{M}^{\mathbf{t-tls}}_{\mathbf{io}:\mathcal{RO}}$	✖	✓✗✗	✖	✓✗✗
		$\mathcal{M}^{\mathbf{u-tls}}_{\mathbf{io}:\mathcal{RO}}$		✔	✔	✓✓✓	✔
		$\mathcal{M}^{\mathbf{u-usb}}_{\mathbf{in}:\mathcal{RO}}$		✔	✔	✓✓✓	✔
		$\mathcal{M}^{\mathbf{u-tls}}_{\mathbf{io}:\mathcal{RW}}$		✓✗✗	✔	✓✓✓	✓✓✓

...

U2F vs g2DT^{dis}

Pros of U2F

- a possibility of privacy
- strong protection against phishing

Cons of U2F

- no feedback
- not independent from the computer

Conclusions

- Detailed threat model for multi-factor authentication protocols
- Analysis of the full system
- Complete automation using PROVERIF and scripts
- Simple, small modifications (adding info to display) that enhance security