Breaking FIDO
Are Exploits in There?
FIDO U2F (Universal 2nd Factor)

- Analyzing FIDO U2F
- Attack and Countermeasures
- Implementation Considerations
- Resources
User Experience

1. Enter username/pwd

2. Insert U2F device

3. Touch U2F device

Sign in

Username or email address
dairnillson

Password [forgot password]

Insert your security key

Press the button on your security key device to finish signing in. If it does not have a button, just re-insert it.

© 2016 Yubico
Core Features & Supporting Sites

- **Scalable**
  - Works across any number of services
  - Remote provisioning

- **Secure**
  - Protects against phishing & MitM
  - Verifies user presence

- **Simple UX**
  - Single gesture operation

- **Open Standard**
  - Native platform/browser support
Challenge Parameter (c)

Client Data = Challenge, Origin, Channel ID

c = SHA-256 hash (Client Data)
Registration

U2F Device

| Generate: | private key | public key | key handle |

Client

| AppID, client data |

Server

| AppID, challenge |

Construct client data containing challenge, origin, channel ID

| pk, kh, sig, client data |

Verify client data
Verify sig
Store pk, kh
Authentication

U2F Device

Retrieve private key
Verify AppID
Sign client data
counter ++

AppID, client data, key handle

Client

AppID, challenge, key handle

Verify AppID

Construct client data containing challenge, origin, channel ID

counter, signature, client data

Server

Fetch public key, key handle for user

counter, signature

Verify client data
Verify sig
Replay sessions

Password DB stolen

Attacker

Attacker

acme.com

Reuse credentials

not_protected.com
Password Reuse & Replay Attack

**U2F Device**

\[ \text{Sign with } k_{\text{priv}} \]

\[ \text{signature(challenge)} \]

\[ s \]

**Client**

\[ \text{challenge} \]

\[ s \]

**Relying Party**

\[ \text{Lookup } k_{\text{pub}} \]

\[ \text{Check signature (s) using } k_{\text{pub}} \]
Attacker site
acm3.com

Phishing (pwd, OTP, SMS, Push)

Attacker

acme.com
Phishing

Attacker site
acm3.com

Proof of user presence

acme.com

Attacker
Phishing Protection

U2F Device

Client

Relying Party

Retriece private key
Verify AppID
Sign client data

challenge, origin, channel id

signature(c)

challenge

c, s

Lookup k_{pub}

Check s using k_{pub}

Verify origin & channel id

© 2016 Yubico
Proof of user presence

1. MitM
   (fake/stolen cert, BEAST)

2. steal assertion

3. use cookie

4. acme.com

Attacker
Hijack User Login Session

U2F Device

Retrieve private key
Verify AppID
Sign client data

AppID, client data, key handle

verify
AppID

Client

Construct client data containing challenge, origin, channel ID

AppID, challenge, key handle

signature

Server

Fetch public key, key handle for user

signature, client data

steal cookie

Verify client data
Verify sig
Channel ID (Token Binding)

- Token Binding (Private Key)
- Token Binding ID: Hash(<Public Key>)
- acme.com
- Token Binding (Public Key)

TLS

- Bind Token with Token Binding ID
Token Binding

Steal assertion

No assertion or cookie reuse

acme.com server

Proof of user presence

Attacker
MitM Protection

U2F Device

Client

Relying Party

ch, o, c

signature(c)

s

c, s

Check s using k_{pub}

Verify origin & channel id

Lookup k_{pub}

Verify AppID

Sign client data

challenge, origin, channel id

challenge
"I promise a user was here"
"The server challenge was: KSDJsdASc8-A17pW"
"The origin was: accounts.acme.com"
"The TLS connection state was: 3454567"

Sincerely, your Browser

- Origin mismatch for key handle => MitM attack!
- Incorrect origin name => MitM attack!
- Channel ID mismatch => MitM attack!
Malware

Attacker

acme.com server
Compromised Client

U2F Device

Retrieve private key
Verify AppID
Sign client data

Client

AppID, challenge, key handle

Verify AppID

signature

signature, client data

Server

Fetch public key, key handle for user

Construct client data containing challenge, origin, channel ID

Verify client data
Verify sig
What are the Alternatives?

- Smart card => Scaling
- OTP/SMS => Phishing
- Push notification => Phishing

Google’s research paper on FIDO U2F deployment: yubico.com/google-study
Implementation Considerations

- **Registration flow**
  - TOFU (Trust On First Use)
  - Attacker enables FIDO on account first
  - User sets up weak backup options

- **Recovery flow**
  - Lowering the level of assurance
  - Lost/stolen authenticator/device
  - Onboarding credentials to new devices
How to Get Started?

Learn

Read the specifications dev.yubi.co/U2F & github.com/dainnilsson/u2f-tutorial
Go through a MiniTwit U2F tutorial MiniTwit training video

Implement

Google reference code github.com/google/u2f-ref-code
Build your own U2F server dev.yubi.co/U2F/libraries
Use Yubico standalone U2F server dev.yubi.co/u2fval

Test

Yubico U2F demo server demo.yubico.com/u2f
Google U2F demo server u2fdemo.appspot.com
Thank you!

Questions