What’s New about Federated Identity Management?

- Federated single sign-on
- Pure browser case. (Else 3-party authentication)
  - Also WS-client versions
  - Also more attributes
- More liability and privacy issues
- Metadata exchange

Scientifically
- Standards
- Management
- Nothing. (Event-based directory integration)
- XML-based. (DSML, SPML, WS-Provisioning)
- More liability and privacy issues

• Also WS-client versions • Also more attributes
State of the Art

- Korman/Rubin 00: Passport *problems*
- Pfitzmann/Waidner 02 etc.: Privacy
- Pfitzmann/Waidner 02, Gross 03: Liberty and SAML *problems*
- Gordon et al: WS protocols, but not FIM
- Gross/Pfitzmann 04: Positive analysis of WSFPI based on “top-down” browser assumptions
- (Gross/Pfitzmann/Sadeghi 05: Detailed browser and user model, reproving “bottom-up”)

Attack Example: SAML Artifact Profile

1a. Authenticate user
2. Redirect to C & artifact
3. GET ... & artifact
4. SAML Request w/ artifact
5. SAML Response: assertion
6. Result page

0. Browse, redirect
A Multi-Layer Vulnerability

1a. Authenticate user
2. Redirect to C & artifact
3. GET ... & artifact
6. Error page with non-SSL link
7a. GET non-SSL page

HTTP Referer: URL w/ artifact

Interrupts channel C↔S
Gets artifact
Impersonates U at C
What Can We Hope to Prove?

- Vulnerable operational environment
  - Based on passwords
  - Fake-screen attacks easy
  - Browser security assumed
  - OS security assumed
- Identity supplier can impersonate user
- Privacy can be good except
  - Not anonymity AND certified attributes
  - Id supplier learns trail of id consumer URIs

Here: Secure channel establishment under appropriate operational assumptions
The WSFPI Protocol – Basis for a Proof

WSFPI ≈ Interop Profile

WS-Fed Passive  WS-Fed Active

WS-Federation

Other WS-Sec*  HTTPS  Tokens
Proof Challenges

- Browsers and users
  - Browser as protocol party – restricted abilities
  - User also a protocol party – zero-footprint browser contains no identity
  - Browser and user might leak “protocol-internal” secrets
- Modularity, e.g., use of secure channels and SAML tokens
- Standard-style presentations
  - We prove rigorous instantiation
WSFPI: Correct Message Flow

0. Browse
4. Redirect(URI$_S$, (wa, wtrealm, [wreply, wctx, wct])
5. Authenticate user
6. POSTForm(a’, (wresult, [wctx]))
7. POST
10. Response

Secure channel

URI$_C$, wtrealm, wreply

name$_S$, URI$_S$, sid$_S$
Machines in Proofs WITH browser model

Claim: Secure channels again
Small Example from the Model

- Behavior of U upon authentication request (critical part to prevent phishing)

\[\text{gui}_{B,U}(\text{request}_{-\text{uauth}}, \text{wid}, \text{host}, \text{sid}, \text{ch}\_type)\]

Waiting → Authentication request

Known trusted server

Failure: missing login / channel insecure
Proceed: uauth successful

\[\exists P \in T_U \mid P.\text{host} = \text{host} \land P.\text{sid} = \text{sid} \land P.\text{login} \neq \epsilon\]

\[\text{ch}\_type \in (\text{uauth}\_sec \cap P.\text{sec})\]

\[\text{gui}_{U,B}(\text{authenticate}, \text{wid}, P.\text{login}, P.\text{sid}, \text{false})\]
Some Remarks on Browser Model

- Channel handling and main HTTP transactions
- User interaction
- Redirect and POSTform for 3-party protocols
- Leakage function, in particular Referer Tag
- Storage and loss of passwords, history, cache

- Proofs need assumptions that unmodeled information leakage really does not occur
  - Usable as future reference for what browsers should NOT do for use in FIM.
First half of B’s state diagram for 1 HTTP transaction

- **User Interaction**
  - Proceed: user denied channel change
  - Failure: wrong window id
  - Local negotiation
  - Channel type changed
  - Secure Channels
  - Insecure Channels
  - Reuse existing channels
  - Connect confirmed
  - Secure channel
  - Channel established
  - Channel reuse
  - New channel required
  - Secure Channels
  - Insecure channels
  - Reusable channel exists

- **Server-cert CA untrusted**
  - Secure Channels
  - Insecure Channels

- **Channel establishment**
  - Secure Channels
  - Insecure Channels

- **Channel reuse**
  - Reusable channel exists

- **First half of B’s state diagram for 1 HTTP transaction**

1. User Interaction
   - Proceed: user denied channel change
   - Failure: wrong window id
2. Local negotiation
   - Proceed: user denied channel change
   - Failure: channel establishment
3. Channel type changed
   - Secure Channels
   - Insecure Channels
4. Connect confirmed
   - Secure Channels
   - Insecure Channels
5. Secure channel
6. Channel established
   - Secure Channels
   - Insecure Channels
7. Channel reuse
   - Reusable channel exists
Summary and Outlook

- **FIM:**
  - 3-party authentication
  - Often with attribute exchange and channel establishment
  - Special: Browser and user instead of dedicated machine

- **Our recent work:**
  - First protocol proof, based on strong assumptions
  - First detailed browser and user models + a lemma

- **Next step:** FIM proof based on browser model
More Information

- Authors:

- Our FIM and WS literature:

- IBM Zurich network security and crypto group: