Automatic Analysis of Browser-based Security Protocols

Avinash Sudhodanan

Alessandro Armando (FBK, coordinator)

Roberto Carbone (FBK, tutor)

Luca Compagna (SAP, tutor)

SECENTIS

A European Industrial Doctorate on Security and Trust

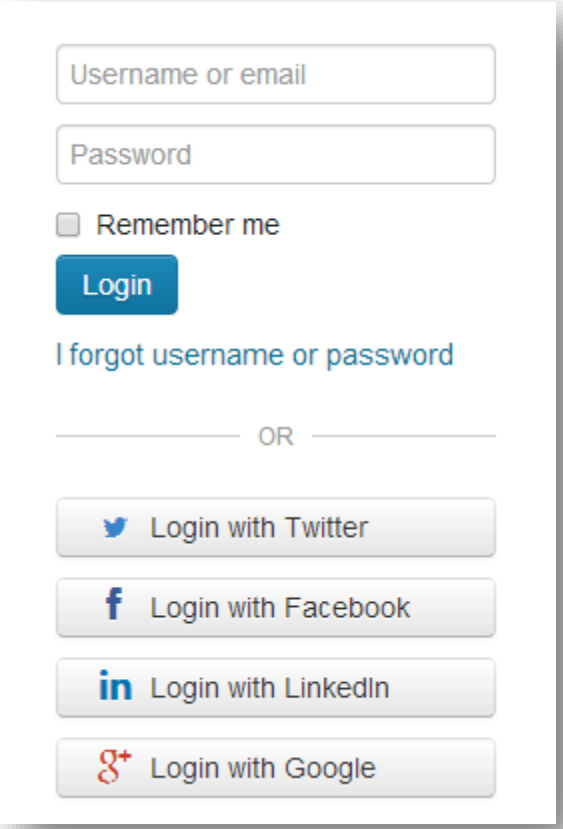
FP7-PEOPLE-2012-ITN

Outline

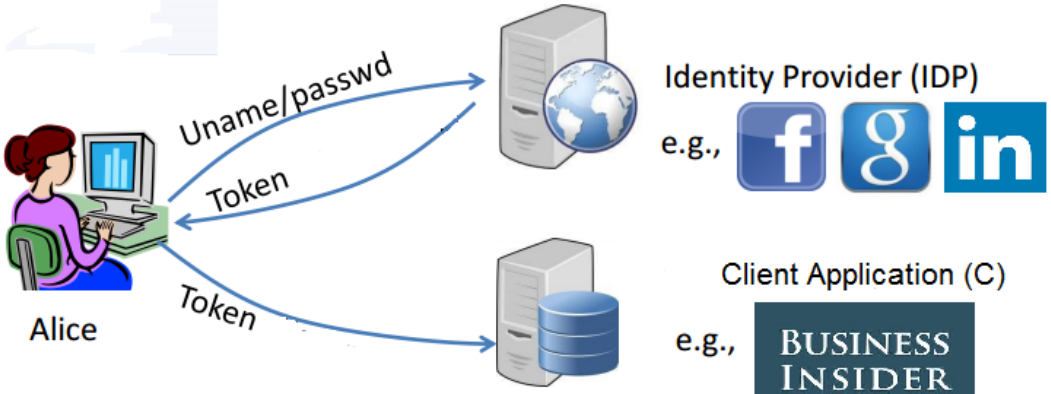
- Context
- Problem Overview
- State of the art
- Proposed Approach
- Conclusion and Future Work

Web Authentication Schemes & Single Sign-On

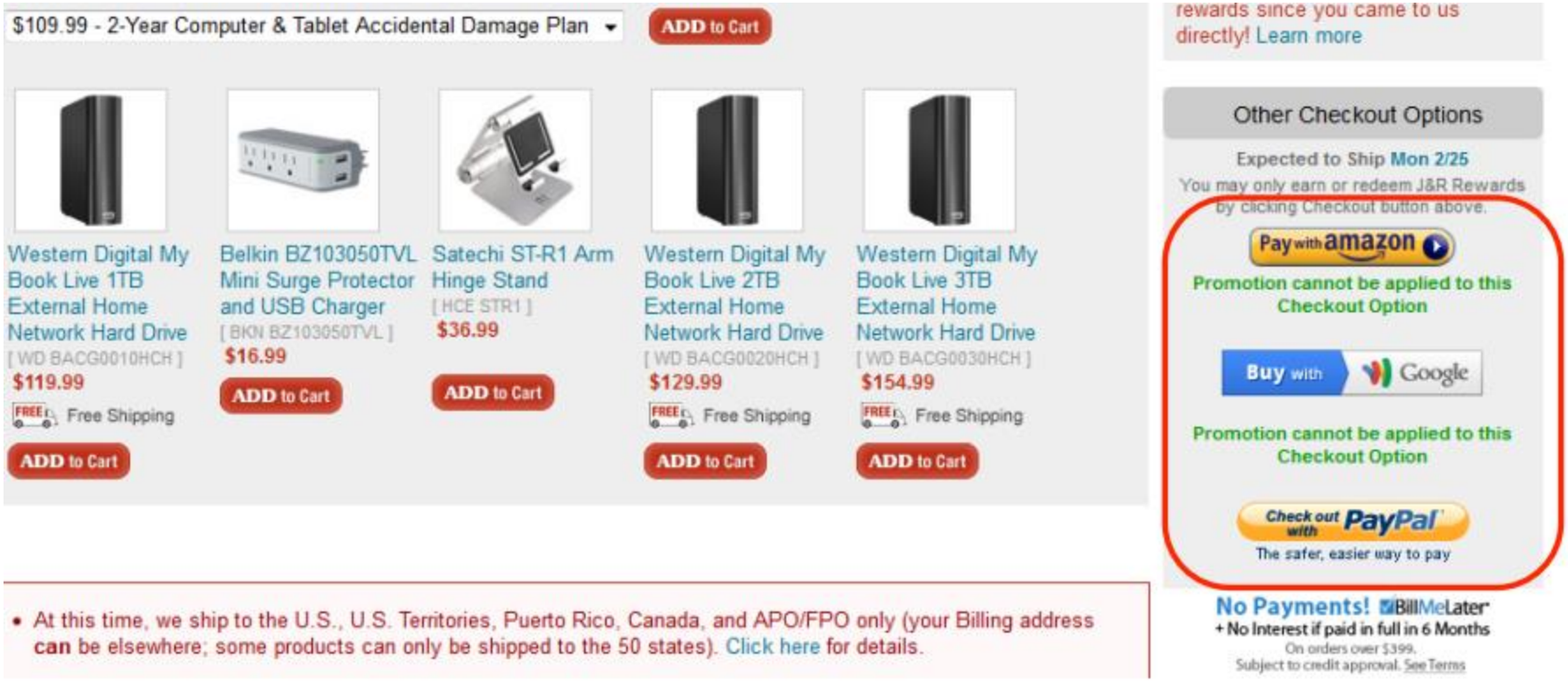
- Web Authentication



- Single Sign-On (SSO)
  - Login with PayPal
  - Sign in with LinkedIn
  - Facebook Login
    - 250+ Million users, 2,000,000 websites
  - OpenID
    - One billion users, 50,000



Integration of third-party Web services



Analysis of Security Protocols

- Current protocol analysis technique: Verification of design-level protocol specification
- But.. security relies on the IMPLEMENTATION

Secure Implementation

- Provide secure implementation guidelines
  - Sign in with LinkedIn

https://developer.linkedin.com/documents/exchange-jsapi-tokens-rest-api-oauth-tokens

FAQ

Q: Why bother with signature validation? What's the point?

A: The main purpose of the credentials_cookie feature is to communicate, securely, two pieces of information from the user's browser to your application: an access_token and LinkedIn member_id for that user.

  - Facebook Login

https://developers.facebook.com/docs/facebook-login/security/

Developers | Apps | Products | Docs | Tools | Support | Search

Security Checklist

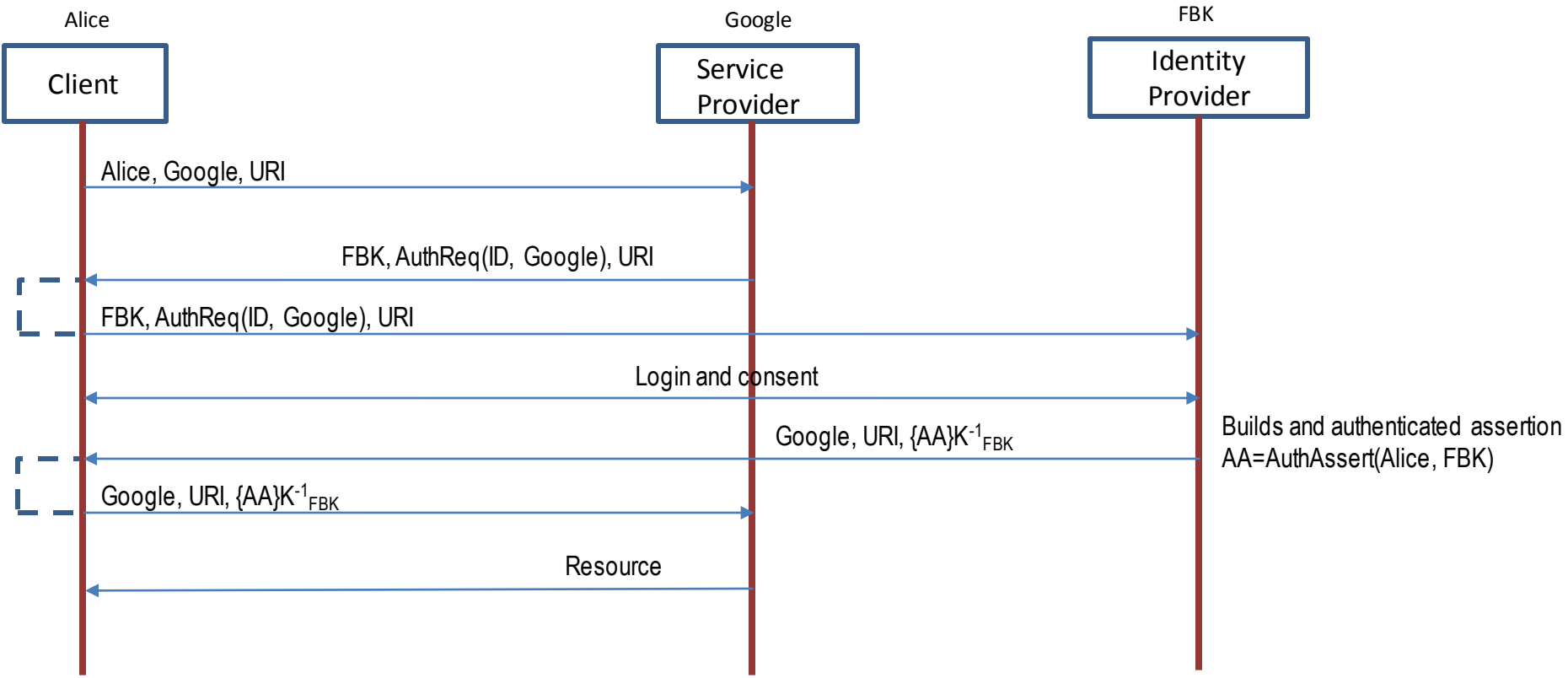
This list below should be considered the absolute minimum that all apps using Facebook Login should implement. Other features will be unique to your app and you will need to always think about how to make your app as secure as possible. Apps that are not secure will lose the trust of their audience and people will stop using them.

- ✓ Never include your App Secret in client-side or decompilable code.
- ✓ Sign all server-to-server Graph API calls with your App Secret.

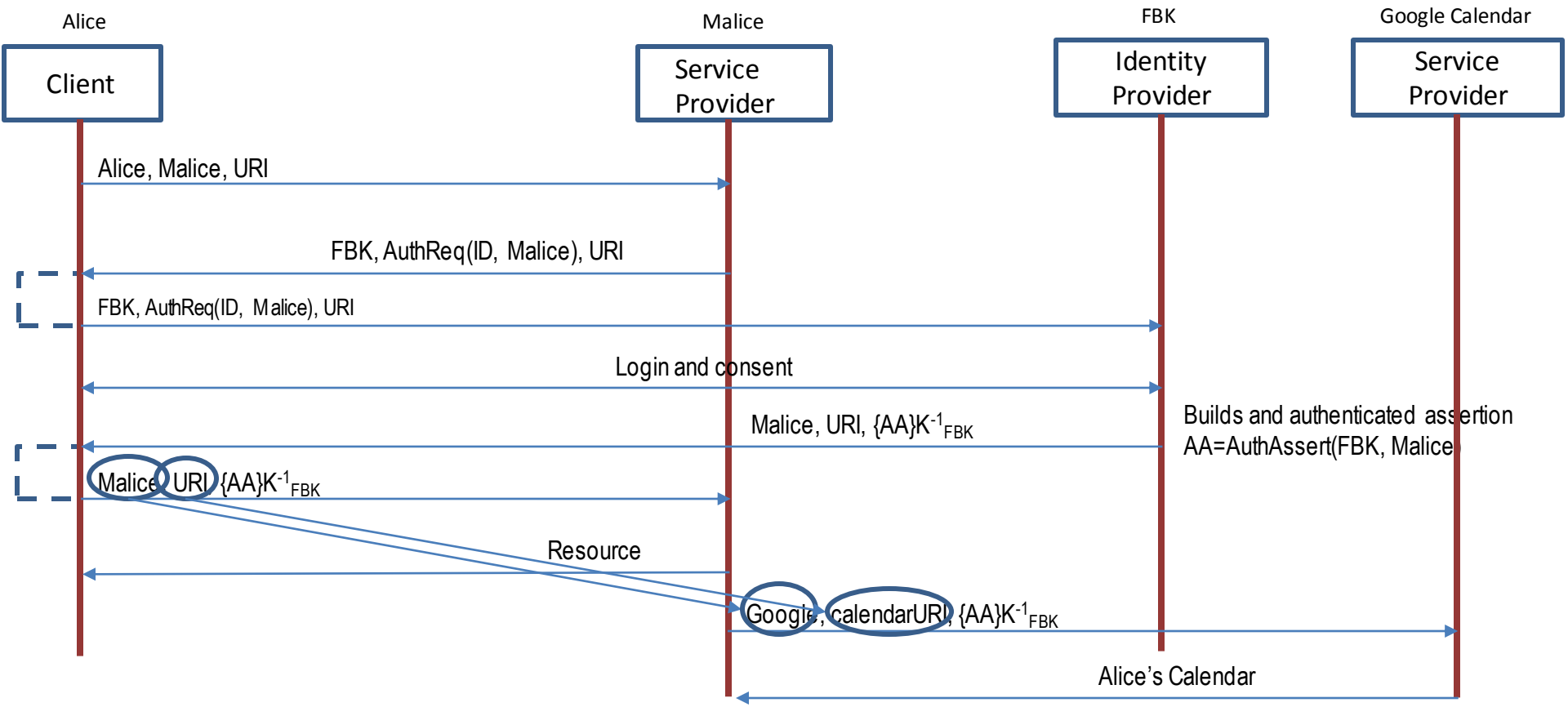
Web Security: Current solutions

- Follow secure implementation guidelines
- Use penetration testing tools (ZAP, Burp, VERA…)
  - Mainly focus on injections vulnerabilities, e.g., XSS, SQLi, …
  - Attack patterns highly dependent on application
  - Logic vulnerabilities out of the scope
- Rely on security knowledge of developer/pen-tester

SAML-based SSO for Google Apps



Attack: SAML-based SSO for Google Apps



State of the art

- **BRM analyzer** [8], **WebSpi**[2], **AuthScan**[3], **SPaCloS(SATMC SAT-based model checker)**[6], **VERA (SPaCloS module)**[15], **WEMM (Giancarlo Pellegrino, Davide Balzarotti)** [5]
  - Good: Evaluates protocol against 3 attacker scenarios and classifies parameters in the communication. Helpful for expert pen-tester
  - Bad: Identifying attack depends on pen-tester's skill
  - Good: Library of ProVerif for modelling Web specific protocols, use power of model checking to discover vulnerabilities
  - Bad: Requires programs to be written in a subset of PHP and Javascript for automatic model extraction
  - Good: Possibility to automatically extract protocol model and test the attack trace discovered by model checker
  - Bad: Difficult to verify the correctness of the model, False positives
  - Good: Nice starting point: combine testing/model checking
  - Bad: Inability to extract the model from the specification
  - Good: Separates attack from attack payloads
  - Bad: Need to manually model the attack sequence
  - Good: Automatically generating test cases for a wide range of modern applications
  - Bad: No provision for adding new attack patterns

Proposed Approach

- Automatically extracting the protocol model from the implementation
  - Extending state of the art techniques
- Applying attack patterns on the extracted protocol model
- Attack patterns that are applicable for wide range of security protocols
- Possibility to add
  - New attack patterns
  - New attack scenarios
- Automatic testing of the implementation

Model Inference: Syntactic Labeling

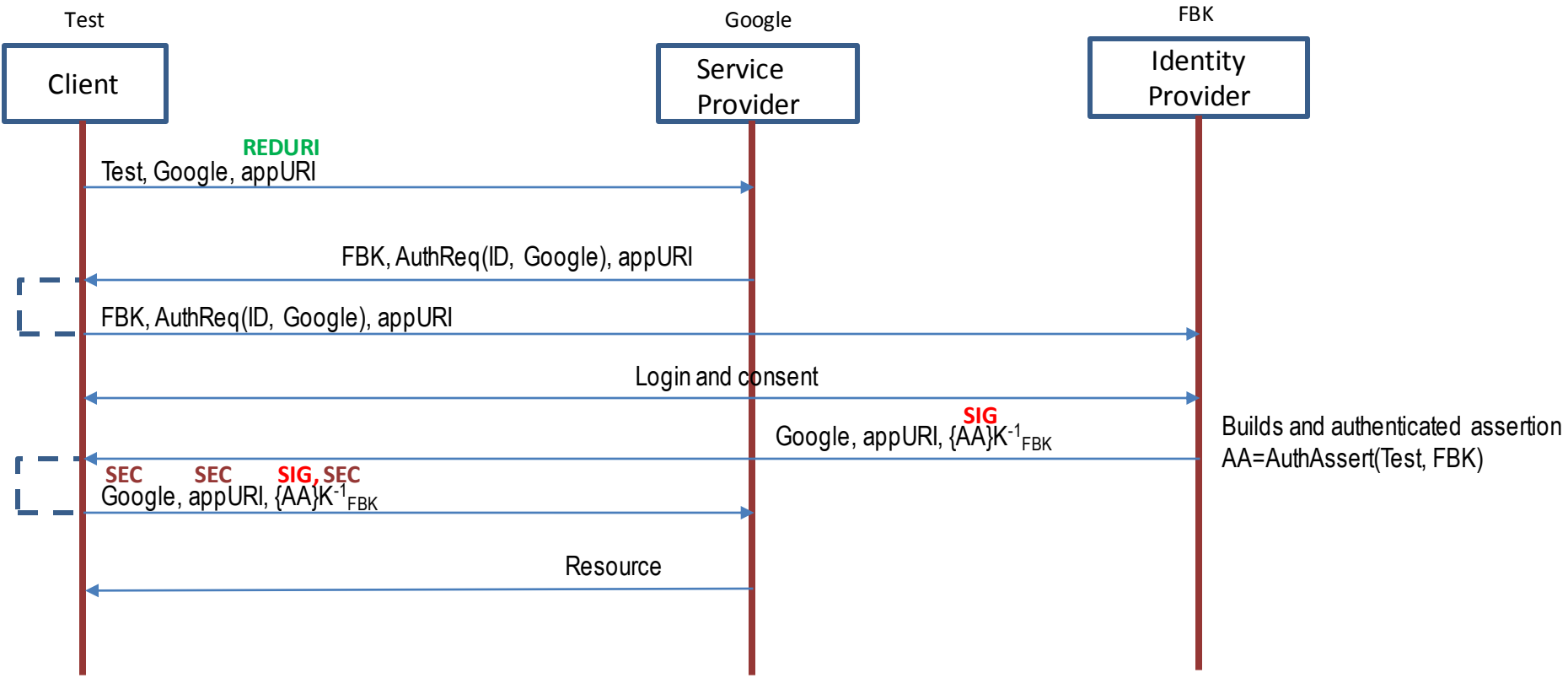
Syntactic Label	Example Value
LIST [8]	*scop=(a,b,c)*
URL [8]	*uri= http://login.google.com*
BLOB [8]	*access_token=e72e16c7e42f292c6912e77*
WORD [8]	*type=code*
UNKNOWN [5]	*#a*
EMAIL [5]	*user_email= example@example.com*
EMPTY	*acope=*
NUMBER	*id=25*
BOOL	*member=True*

Model Inference: Semantic Label

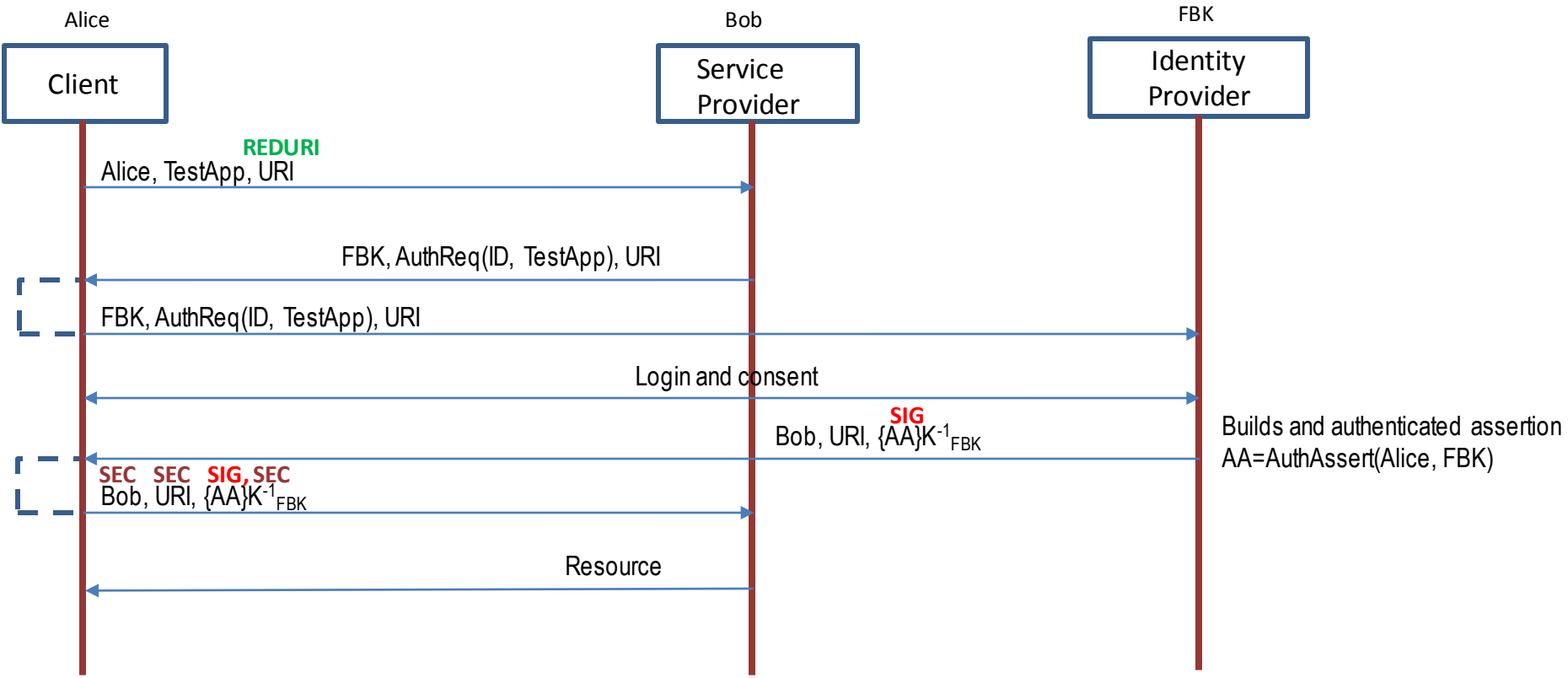
Label	User 1, Application 1	User 1, Application 2	User 2, Application 1	User 2, Application 2
UU (user-unique) [8]	A	A	B	B
SU(session-unique) [8]	A	B	C	D
App Unique (AU)	A	B	A	B

Label	Description
SEC (secret) [8]	Parameter is necessary for the authentication
SIG (signature) [8]	Signature
BG (browser-generated) [8]	Element present in a request but not included in preceding responses
REDURI(redirection url)	URL which was passed as a request parameter and later found in the Location header of a redirection response

User: Test, Application: GoogleApp

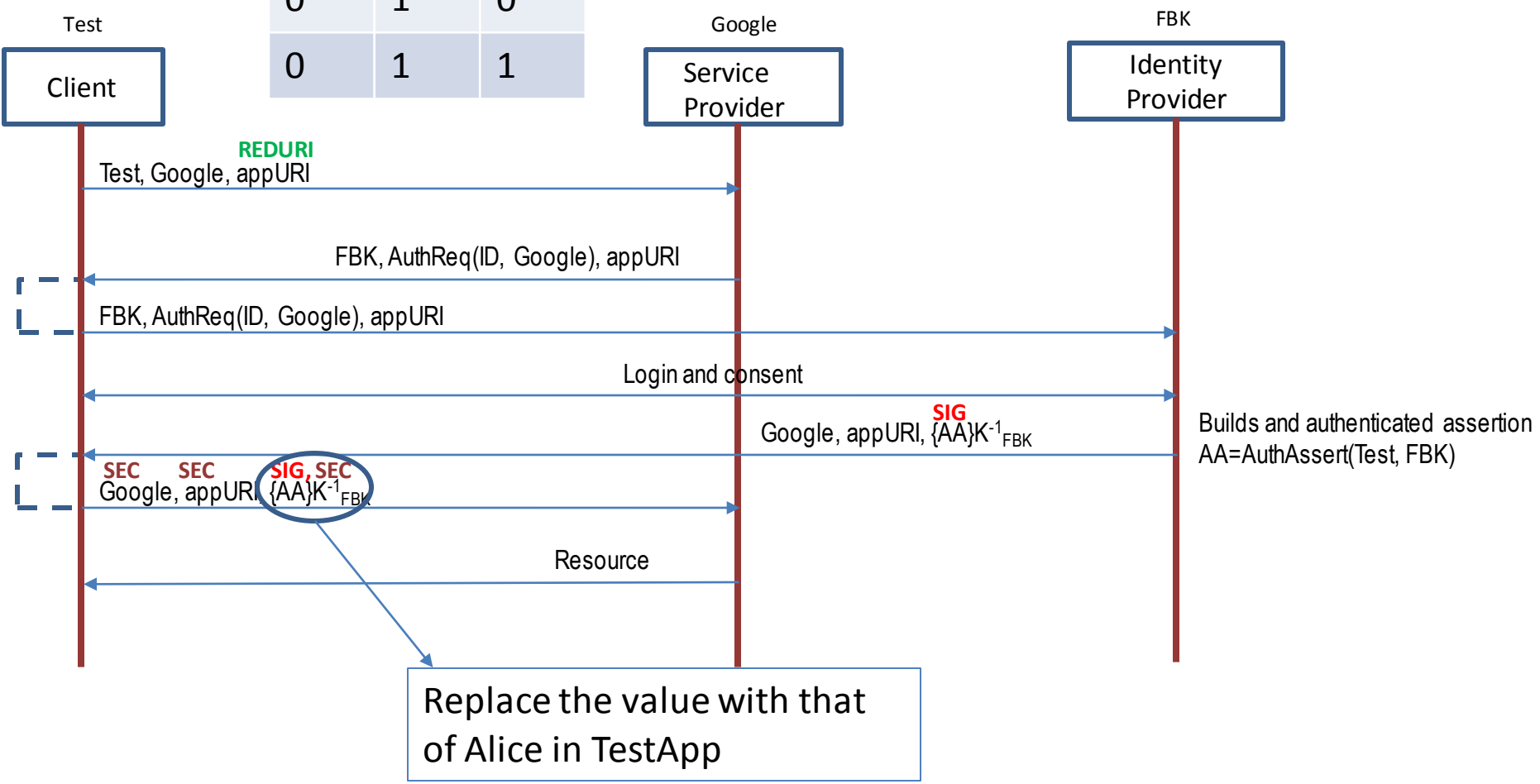


User: Alice, Application: TestApp



Attack Pattern: User-Test, Application-GoogleApp

Idp	url	sign
0	0	1
0	1	0
0	1	1



Attack Pattern: Replay attack

- **Goal:** Replay session parameters in order to gain unauthorized access to at least one User Unique element in U1C1
- **Preconditions:** There is at least one element with semantic type as SEC in U2C1
- **Actions:**

  AND 1. Initialize test with baseurl of U2C1 & useractions of U2C1

  2. Set variable sec_list as all elements in U2C1 that has semantic type as SEC

  3. Start executing test

  AND 3.1. For each combination of elements in sec_list, replace their value in the Requests of test with corresponding value in U1C2
- **Post conditions:** There are elements of U1C1 with semantic type as User Unique & origin as responsebody in trace of test

Conclusions

- Existing testing methods are insufficient for automatically testing security protocols
- We discovered a number of security issues in the implementation of widely used SSO protocols (LinkedIn, Yahoo)
- We propose a system that can automatically generate test cases for evaluating the security of protocol implementations
  - Current status: Identifying design patterns for representing protocol, attacks and threat model

Future Work

- Refine the proposed approach and provide a prototype of the tool
- Testing security protocol implementations
- Integrate with a legacy penetration testing tool
- Application of model checking for improving the effectiveness of the vulnerability detection technique

References 1/3

- [1] Devdatta Akhawe , Adam Barth , Peifung E. Lam , John Mitchell , Dawn Song, “Towards a Formal Foundation of Web Security”, Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium, p.290-304, July 17-19, 2010
- [2] Bai, G., Meng, G., Lei, J., Venkatraman, S. S., Saxena, P., Sun, J., et al. (2013). “AuthScan: Automatic extraction of Web authentication protocols from implementations.” In Proceedings of the 20th annual network and distributed system security symposium.
- [3] Chetan Bansal, Karthikeyan Bhargavan, Sergio Maffeis, “Discovering Concrete Attacks on Website Authorization by Formal Analysis”, Proceedings of the 2012 IEEE 25th Computer Security Foundations Symposium, p.247-262, June 25-27, 2012.
- [4] Rui Wang, Yuchen Zhou, Shuo Chen, Shaz Qadeer, David Evans, and Yuri Gurevich (2013, August). “Explicating SDKs: Uncovering Assumptions Underlying Secure Authentication and Authorization.” In Proceedings of the 22th conference on Security symposium

References 2/3

- [5] Giancarlo Pellegrino and Davide Balzarotti. “Toward Black- Box Detection of Logic Flaws in Web Applications.” Network and Distributed System Security (NDSS) Symposium, 2014
- [6] http://www.spacios.eu/
- [7] http://prosecco.gforge.inria.fr/personal/bblanche/proverif/
- [8] Rui Wang, Shuo Chen, XiaoFeng Wang, “Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services,” Proceedings of the 2012 IEEE Symposium on Security and Privacy, p.365-379, May 20-25, 2012.
- [9] Rui Wang, Shuo Chen, XiaoFeng Wang, Shaz Qadeer, “How to Shop for Free Online -- Security Analysis of Cashier-as-a-Service Based Web Stores”, Proceedings of the 2011 IEEE Symposium on Security and Privacy, p.465-480, May 22-25, 2011.
- [10] G. Lowe. “A Hierarchy of Authentication Specifications.” In CSFW, pages 31–43, 1997.

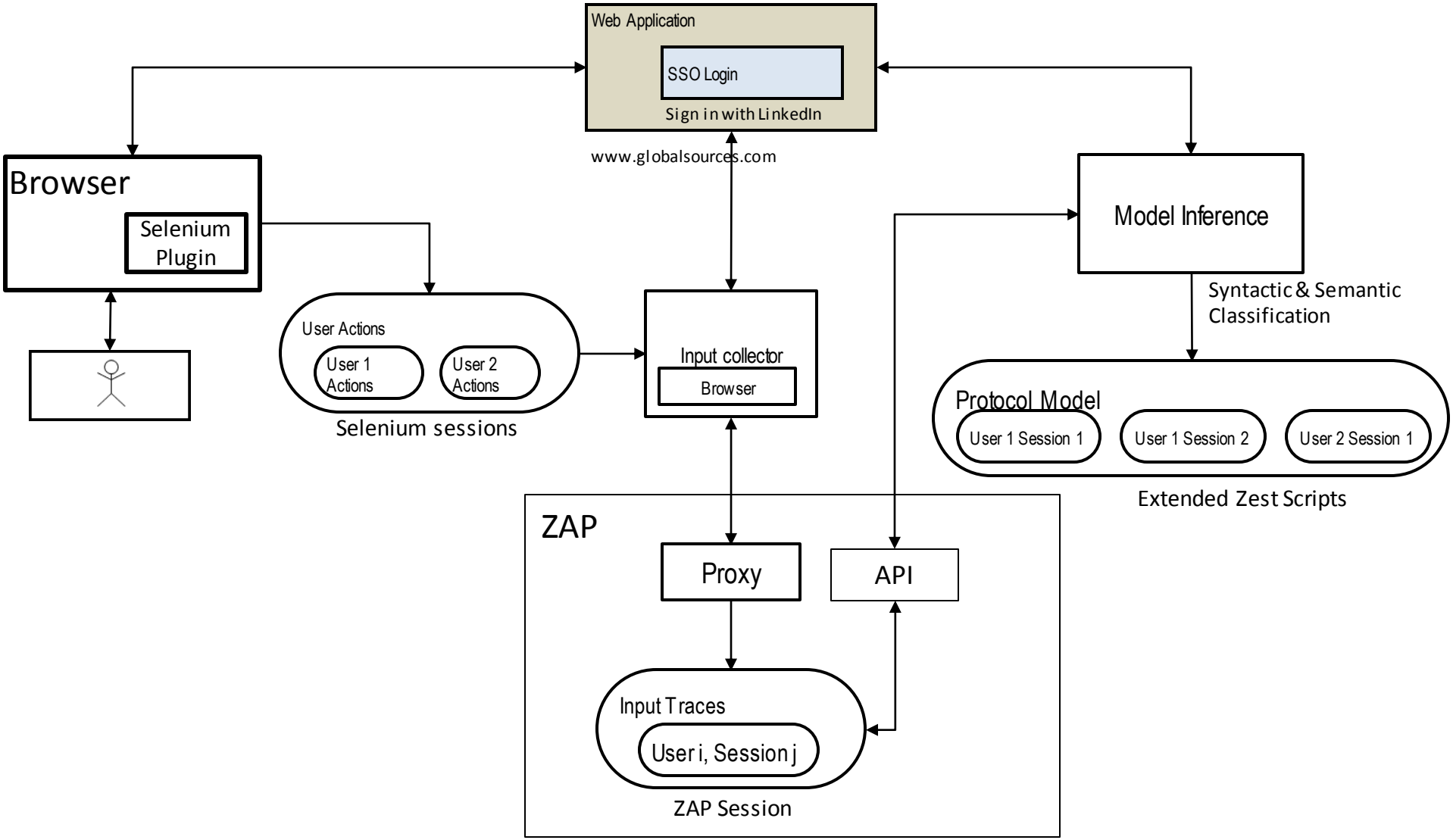
References 3/3

- [11] T. Y. C. Woo and S. S. Lam. “A Semantic Model for Authentication Protocols.” In S&P, pages 178–194, 1993.
- [12] A. Armando, R. Carbone, and L. Compagna, “Ltl model checking for security protocols,” in Computer Security Foundations Symposium, 2007. CSF ’07. 20th IEEE, July 2007, pp. 385–396.
- [13] Zhou, Yuchen, and David Evans. "SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities." http://www.ssoscan.org/
- [14] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
- [15] VERA: A Flexible Model-Based Vulnerability Testing Tool
- [16] https://github.com/mozilla/zest/wiki

Thank You

sudhodanan@fbk.eu

Architecture Diagram



Architecture Diagram cont.