Increasing Security of WebIDs Through Biometrics

Tony Gwyn
Winston Salem State University
Winston Salem, NC
tgwyn482@rams.wssu.edu

Taylor Martin
North Carolina A&T State University
Greensboro, NC
tsmarti1@aggies.ncat.edu

Dr. Albert Esterline
North Carolina A&T State University
Greensboro, NC
esterlin@ncat.edu

SoutheastCon 2020 | 978-1-7281-6861-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/SoutheastCon44009.2020.9368274

Abstract— **We are creating a streamlined way to adapt WebIDs [1], and biometrics [2] to the cyber world. This involves building a user authentication system that enables quick, fast and secure access. It is understood that compared to traditional username and password user authentication, WebIDs are designed to provide such services. Nevertheless, if an intruder either has direct access to the user's computer or somehow gets the unique certificate of the user, important information can be stolen with solely the use of WebIDs. Since biometric data (e.g. fingerprints, iris scanning, etc.) is unique and not easily duplicated, this possibility can be avoided by including biometrics in the authentication process. We also include an enrollment protocol that checks whether a user has a WebID while trying to access a server. If they do, we allow the user access to the server, and if they do not, by accessing their own server, we register the user for a WebID with their permission. Implementing these features in the WebID protocol will greatly enhance user authentication safety.**

Keywords—WebID, authentication, biometrics

I. INTRODUCTION

Using usernames and passwords is the current standard method for authenticating accounts. Nevertheless, a number of issues can arise with this approach. It can be a drawback for a multi-account client. Remembering a multitude of different passwords for different websites may be challenging for one user but deciding to write down or record passwords can lessen security. Another concern is that during transmission, usernames and passwords can be intercepted. It is possible for hackers to make brute force attempts to guess one's username and password. The alternative to multiple passwords is to use a single password for multiple accounts, but if one password is compromised, this can pose a serious threat. The use of WebIDs [1] is a solution to this problem. With the WebID protocol, it is possible to remove the complexity of recalling usernames and passwords and store all user information on the user's computer. Only by verifying a user's private ID will this protocol allow access to a server. While this is nice, it may lead to other important issues. As far as WebIDs are concerned, anyone in possession of the user's device also has access to all the user's accounts and information. With that being said, if the device were to be compromised by anyone, then all of the user's information can be easily accessed and stolen. A two-factor approach through biometrics and WebIDs would be a solution to this problem. Using these two at the same time, security is greatly increased by only allowing access to a server if both the WebID certificate and the verified biometric match is confirmed. Several extensions of the WebID protocol have been reported in the literature, [16] describes an extension that gives it the ability to manage groups of agents and control their access to resources on the Web, while [17] proposes an extension to the WebID protocol that allows for delegation of access authorization without compromising privacy. This paper describes a cleaner version of our two-factor approach through biometrics and WebIDs [14] and adds the enrollment protocol. To our knowledge, our work alone enhances WebID with biometrics.

This paper discusses the WebID protocol incorporating biometrics and an enrollment protocol that has simplified identity management and allows a single sign-on. Section II identifies two biometric techniques, one of which is our current WebID enhancement, and the other is a technique developed by our colleagues and will be integrated into a later version of our enhancement. Section III introduces the Semantic Web, as well as what it does, and how it relates to our work. This section also presents both URIs and CoolURIs, as well as the Resource Description Framework (RDF). Section IV discusses the WebID protocol and gives a brief overview of what it is and what it does Section V introduces our protocol for WebID+Biometrics. As WebID takes a decentralized approach to identity management and the use of biometrics involves non-trivial processing, the creation of a WebID enrollment protocol poses some challenges; this enrollment protocol is described in Section VI discusses our implementation using MEAN stack components and software to detect objects in real time. This section also includes our problems faced, and a detailed description of our enrollment process is included. Section VIII introduces and defines Solid, giving background about

its application. Section IX concludes and addresses future work, which could be important due to the possibilities posed by the way WebID uses a profile document and the integration of WebID into vast-scale innovative projects.

II. Biometrics

Biometrics is a technique that accepts a user's physical characteristics to identify them. Biometrics is the process of identifying, verifying or recognizing a living person based off their physical or behavioral characteristics [2]. For biometric verification, we use two different protocols. One is the Local Binary Pattern (LBP)[3], which is less complex and is currently being implemented in the protocol. The other is Genetic and Evolutionary Feature Extraction (GEFE) [4]. We have a stand-alone version of GEFE, which we plan to incorporate into the WebID+Biometrics protocol in the future.

A. LBP

LBP is a texture operator that marks an image's pixels by thresholding each pixel's neighborhood. It then sees the result as a binary number. For each patch in an image, there are 256 different bins. The user can choose the number of patches. Using a 3x3 graph that focuses on the center area, each pixel in each patch is compared to its neighbors. Compared to all its surrounding regions, the center area is used to assess whether the pixel intensities are higher or lower. When the number is equal to or greater than the center region, a 1 will represent the region. If a region around it is less than the center region, it is defined by a 0. Then we use the 0's and 1's around the center to construct a binary number between 0000000 and 11111111, which is translated to a decimal number between 0 and 255. For that patch, each decimal number is applied to a sub histogram. Then all the sub-histograms the of patch are concatenated in order to create a final vector for comparison. The Manhattan distance metric is used for feature matching.

B. GEFE

Genetic and Evolutionary Feature Extraction (GEFE) is an evolutionary feature extraction technique, which can be used in conjunction with LBP to evolve unique Feature Extractors (FE) for biometric-based authentication systems. GEFE is part of genetic and evolutionary computation (GEC), [4] is generally used for optimization. Multiple FEs are used and tailored to the system to reduce the error rate in the biometric system. This also helps to increase the system's overall security against replay attacks, which are attempts to fraudulently use valid data in a way which was not originally intended. The GEFE protocol is virtually immune to these replay attacks, rendering them highly ineffective [4].

III. Semantic Web

The Semantic Web can be described as a decentralized information space for sharing machine-readable data, at very low integration costs. This information is expressed as statements about resources; those resources are identified by Uniform Resource Identifiers (URIs), which themselves are at the heart of the Resource Description Framework (RDF) [4]. While URIs originally referred to just documents, URIs are now often used to refer to either logical or physical resources, such as abstract ideas or real-world objects, respectively. [5]. CoolURIs [5] specifically denote a resource, but then dereferences to a document about that resource by either using 303 redirection, or fragment identifiers. RDF information is represented as triples: specifically, subject, predicate, and object. The subject identifies the resources, whereas the predicate denotes the traits of the subject, or describes the relationship between the subject and the object. Lastly, the object is the value for the predicate of the resource to which the subject relates.

NSF Research Experiences for Undergraduates, Grant No. 1460864 funded this research.

IV. Introduction to WebIDs

A WebID is a Cool URI that denotes an entity such as a person, organization, group or device. It is in the subject-controlled address space so does not overlap with other identifiers in any system. Certificate authorities need a form of authentication based on centralized systems. It ensures that, for each service they use, a customer must have multiple accounts and identifiers. A new register must be established for each service, which can be a burden on the user as well as the service [6]. With the WebID protocol, a WebID dereferences to a user's WebID profile, which contains structured data in RDF using the FOAF (Friend Of A Friend) ontology [7]. The profile essentially contains a FOAF graph some of whose triples link the subject to their friends through foaf:knows relationships; other triples provide the subject's attributes such as their name.

V. WebIDs + Biometrics

WebIDs offered an easy and safe alternative to traditional user / password authentication. Unfortunately, if an attacker gains direct access to a user's computer or if the user's unique certificate is stolen, their personal information can still be compromised. The addition of biometrics to the authentication process solves the problem as biometric data (e.g., fingerprints, iris scanning, etc.) is distinctive and not easily reproduced. If a biometric element can be added to WebID profiles, both their WebID and biometric authentication can be used to verify users. We have introduced a simple user validation process, which is widely applicable through the internet, and protects against intrusion.

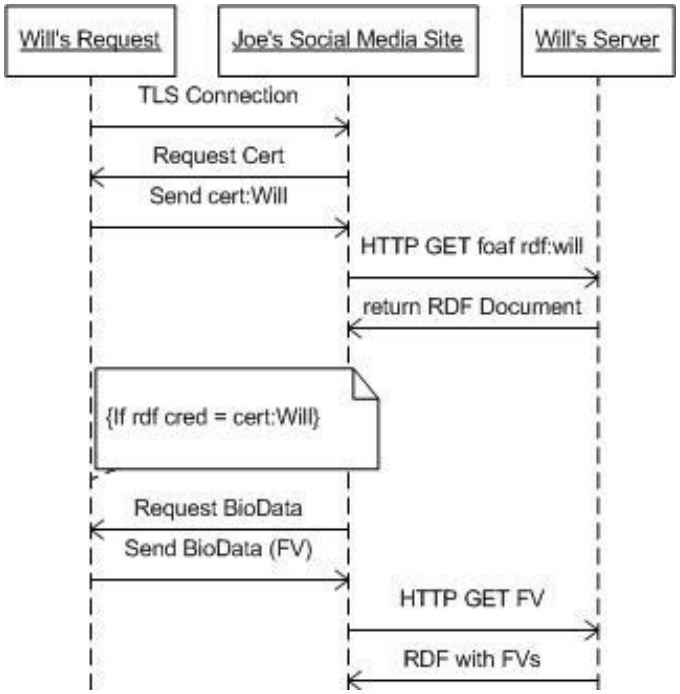

Fig. 1. Sequence Diagram

As an example of how the protocol works, refer to Fig.1. There are two users, Joseph and William. William wants access to Joseph's social networking service. William's client starts by sending a TLS request to Joseph's social networking service; Joseph's social site then sends a request for a certificate to William's client. The client sends the certificate over in return. After the certificate is received, Joseph's site pulls the URI of William's profile (on William's server) and de-reference it. Joseph's site compares modulus and exponent in William's profile against the modulus and exponent from the client. If they match, then Josephs' service sends the RDF for the specific FE that has been randomly generated. William's client provides the Feature Vectors (FVs) in RDF for a biometric artifact requested. Joseph's social site de-references a document on William's server that contains his biometric information. Joseph's service verifies that William's biometric matches the previous enrolled FV. Lastly, if verification matches, then Joseph's service notifies William's client of access. Finally, if verification results in a match, then Josephs service will notify William's client that he may access the social networking site [14].

VI. Federated Biometric Enrollment

A person generally uses biometrics to enter a system for two reasons, either to allow access to it or to enroll in the system. Usually, a person will provide additional information about themselves in addition to their biometrics in the case of traditional enrollment. Some name will be captured and compared to the biometric enrolled. The enrolled biometric may be changed either by the individual using another password or an identifier unique to that individual. Our enrollment protocol is used when users are trying to access a server. The protocol ensures that the user has a WebID before proceeding. If the user does not have a personal WebID, the protocol then directs them to begin the process of setting one up.

VII. Implementation

Node.js, Express and Angular4 were used to implement our protocol. A simple http server was developed to provide routing using the Node and Express library. Angular and TypeScript have been used to implement the application on the client side, which does a lot of computational work. On the client side, the application is delivered via HTTP to the client's browser, which communicates with the server application via a simple REST API [8] to handle the WebID authentication protocol. The server requires a valid WebID x.509 certificate to be issued by the user, which can then be checked by the server. If the user does not have a current WebID, the Enrollment protocol steps in to handle setting a WebID certificate up for the user. Upon confirmation or rejection of a WebID certificate, the session of the user is updated to reflect this. To validate the WebID, the Alt-name subject field of the certificate is dereferenced to obtain the correct URI for the WebID profile. The original certificate's key will be compared with the public key in the profile document [14].

Our goal was to integrate WebID and biometrics effectively, with the goal of having a client verification technique that is efficient, typically appropriate, and safe against interference. The first move was to create a simple client capable of capturing video and taking screenshots using Angular, a JavaScript framework that makes it easier to construct web applications. We started to gain access to the webcam of the client. To do this, the HTML5 Canvas element was first used to draw the webcam's photo. We got the entire picture data of the image, which included RGB values for each pixel. When we had the RGB data for every pixel, we were able to change the picture to a grayscale picture. We used js-objectdetect to identify the face on the canvas. We could identify the face on the Canvas using js-objectdetect [9], a JavaScript library for real-time object detection. From that point on, we found the contours of the face; then, using methods provided by Canvas, we manipulated the image and used LBP to detect the characteristics and facial similarities. Now that we have the face data, it can be stored in a server database along with the user's information for later comparison upon login.

As for the enrollment protocol, the user will begin on the "Homepage" of the server they are trying to access. As of now, we prompt the user and ask if they currently have a WebID. If the user denies the registration will be sent to the user. As mentioned above, a certificate is created for the users

' personal server. The user is encouraged to take a photo for the collection of biometric data, which will then be stored in a database on the server that the user is attempting to access. The access process continues if the user acknowledges that they have a WebID. Once the user tries to access this server and their WebID certificate is confirmed, the user will go through the face capture process again and have their current face data compared with the server's database face data.

VIII. SOLID

Solid is a proposed set of conventions and tools for building decentralized Web applications. Led by the 'father' of the Web, Tim Berners-Lee, Solid is an acronym, derived from "social linked data" [10]. The solid technology is based on existing W3C standards and protocols, including the WebID protocol, which plays a central role. Following WebID, an identity is bundled with its authentication credentials, and identity, resources and containers are URI addressable. Data consists of resources and nestable containers (like file folders), and the user controls access. Solid is based on Linked Data principles and is modular and extensible. Solid seeks decentralized identity, storage and authentication, with user control of authorization.

A community of contributors allows for Solid to expand and grow, keeping up to date with any new standards and protocols. Solid is a multitude of things, wrapped into one easy to learn and configure package. It is a tech stack, which is a set of standards and vocabularies that are complementary, and provide capabilities such as identity, authentication and login, authorization and permission lists, and much more. Solid is also a specification document which describes a Rest API that extends the previously mentioned standards. Solid is a set of servers, which implement this specification, as well as an ecosystem of apps, identity providers and helper libraries. Finally, Solid is a community; a large group helping to provide discussion, documentation, and presentations [10].

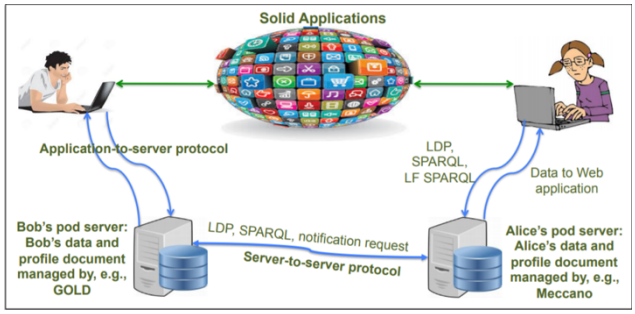

Fig. 2. SOLID Diagram

Fig. 2 gives an overview of the Solid platform [11]. User data is stored in a personal online pod that resides in a server. The user has full control over their identity, by the use of an RDF profile document. The user loads the Solid application from an app provider, and then that application searches the identity profile to obtain the user's 'pod'. Using the user's profile, the application is then able to follow web links to discover data on the user, performing authentication and verification as needed [11]. It should also be noted that with Solid, users can have more than one 'pod', from different providers, and at the same time users can easily switch between those providers [12]. This flexibility allows for users to have full control over their web data. We intend to follow this project (we are a member of the interest group), and we will look for where our effort may have an impact on this potentially revolutionary technology.

IX. CONCLUSION

This project attempted to build a user authentication and authorization system that would enable fast and secure interactions for the cyber world. The best way to accomplish this was to use WebIDs and biometrics to both streamline and safeguard the process. WebIDs are vastly superior to typical username and password authentication; however, important information can still be compromised if an unauthorized intruder gains access to the user's workstation or their unique certificate. The use of biometrics helps to alleviate this problem, as biometrics are both unique to each user, and not easily spoofed or duplicated. The created enrollment protocol was successful in allowing authorized and enrolled users with WebIDs to access the server, while inviting users without a WebID to register for one, if they so choose. The work reported here is (to the best of our knowledge) the only work that enhances the WebID protocol with biometrics.

As for future work we plan to eventually develop some form of two-factor authentication with another device so that users can access their personal WebID certificates on different devices. We also plan to implement Machine Learning / Neural Network technology [15] for a much better facial feature matching system.

ACKNOWLEDGMENT

This research is based upon work supported by the National Science Foundation (Award # 1900187).

REFERENCES

[1] WebID. (n.d.). Retrieved from https://www.w3.org/wiki/WebID.

[2] Sabol, C., Nick, W., Earl, M., Shelton, J., and Esterline, A. 2009. "The WebID Protocol Enhanced With Group Access, Biometrics, and Access Policies." In MAICS 2016.

[3] Pietikäinen, M. (n.d.). Local Binary Patterns. Retrieved from http://www.scholarpedia.org/article/Local_Binary_Patterns.

[4] Nick, W., Shelton, J., Sabol, C., and Esterline, A. “Federated Protocol for Biometric Authentication and Access Control,” In 2017 Computing Conference , 18-20 July 2017, London, UK.

[5] Sauermann, L., and Cyganiak, R. (Eds.), Cool URIs for the Semantic Web ,W3C,2008, available online at http://www.w3.org/TR/cooluris/.

[6] Story, H., Harbulot, B., Jacobi, I., and Jones, M. 2009. Foaf+ ssl: “Restful Authentication for the Social Web”. In Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009).

[7] Brickley, D., and Miller, L. “FOAF Vocabulary Specification 0.99,” W3C, 2014, available online at http://xmlns.com/foaf/spec/.

[8] REST API Tutorial. (n.d.). Retrieved from https://restfulapi.net/.

[9] js-objectdetect. (n.d.). Retrieved from https://www.npmjs.com/package/js-objectdetect.

[10] Solid GitHub, (“Solid - Re-decentralizing the web (project directory)”, online at https://github.com/solid/solid, accessed 12/18/2019.

[11] Mansour, E., Sambra, A., Hawke, S., et. al. “A demonstration of the solid platform for social web applications”, in Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 223-226.

[12] Sambra, A., Hawke, S., Zereba, M., et. al. “Solid: A platform for decentralized social applications based on linked data”, Technical Report, MIT CSAIL and Qatar Computing Research Institute 2016, available online at http:/emansour.com/research/lusail/solid_protocols.pdf.

[13] Ahonen, T., Hadid, A., and Pietikainen, M. “Face Description with Local Binary Patterns: Application to Face Recognition’”, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28, Issue 12, pp. 2037-2041, December 2006.

[14] Martin, T., Zhang, J., Nick, W., Sabol, C., and Esterline, A. (2018, March). Implementing webIDs + biometrics. Proc. ACMSE 2018 Conference (ACMSE '18).

[15] Le, J. (n.d.). A Gentle Introduction to Neural Networks for Machine Learning. Retrieved from https://www.codementor.io/@james_aka_yale/a-gentle-introduction-to-neural-networks-for-machine-learning-hkijvz7lp.

[16] C. Sabol, W. Nick, M. Earl, J. Shelton, and A. Esterline (2016, April). The WebID Protocol Enhanced with Group Access, Biometrics, and Access Policies. Proc. 27 th Modern Artificial Intelligence and Cognitive Science Conference (MAICS 2016), pages 89–95.

[17] Sebastian Tramp, Henry Story, Andrei Vlad Sambra, Philipp Frischmuth, Michael Martin, and Sören Auer (2012, Nov.). Extending the WebID protocol with access delegation. Proc. 3 rd International Conference on Consuming Linked Data (COLD'12), pages 99–111.