Multifactor Transparent Authentication

Alcides Rivarola[a1], Cristian Cappo[a]
[a]Facultad Politécnica, Universidad Nacional de Asunción
Received on August 15, 2014 / accepted on *****, 2014

Abstract

From the beginning of the Internet, web applications authentication is a real concern issue for users that need to protect personal or very sensitive information. The traditional authentication mechanism, username and password, although it may have potential vulnerabilities, till now remains the most widely used authentication method on the web.

We present the Multifactor Transparent Authentication (MFTA), a scheme that aims to increase the security of web applications authentication compared to a two factor authentication mechanism, but keeping the ease of use and the deployability challenge. It uses the user's PC as a second factor device, and checking some features of the user's PC, in the background, when authenticating. It focuses mainly in organization where the users do not need to change his computer, or do not have the need to connect from another computer.

Keywords: Authentication, Multifactor, Login, Web, Security.

1 Introduction

The username and password till now remains the most widely used authentication mechanism on web applications. It is easy for developers to implement on a server and the users are used to it.

There are many problems with this mechanism. One of them could be the reuse of passwords by users on across many web sites [3], so the compromise of one web site could leads to the compromise of others web sites. According to study conducted by Internet security company BitDefender, over 250,000 usernames, email addresses and passwords, it found that 75% of users had one common password for social networking and accessing their email [7,11]. For those users who use a different password for each site, the problem is to remember all of them when there are a significant amount of sites [10].

[1]E-mail Corresponding Author: aerivarola@gmail.com

Another problem is phishing, a fake page of the original web site, in which users enter their credentials and through which identity theft occurs [8]. These threats cause mainly financial damage, but they can also cause damage ranging from minor annoyances to real threats to life.

There are many alternatives to the traditional username and password authentication mechanism. The two factor authentication [1], the Federated login and password managers are good approaches but Bonneau evaluation framework [4] shows that none of them meets the requirement of security, usability, or deployability in practice. It is known that most of the users prefers comfort over security when it comes to authentication mechanism, thus usability is a very important issue on authentication. The schemes described before change somehow the user experience so it makes it difficult to implement within an organization.

For an organization that has a web application with valuable information, and just specific clients need to access from inside or outside the organization, none of the existing authentication mechanisms provide enough security without affecting the usability and deployability, for that we propose the Multi Factor Transparent Authentication scheme (MFTA) which tries to deal with the issues mentioned above.

This paper is organizated as follows. Section 2 presents methods review in authentication. Section 3 describes our proposed scheme. The architecture of MFTA is presented in the Section 4. We evaluate the MFTA scheme considering the Bonneau evaluation framework [4] and it is presented in Section 5. Finally, Section 6 presents the final considerations, conclusions and our work in progress.

2 Related work

In this section we examine some related work and how they attempt to address the security issues with the authentication.

The Federated Login, like OpenID [2], Facebook Connect [9] and BrowserID [14], allows users to have just one account on an identity provider to which the websites consult for the validity of the user's identity. All other websites (usually called relying parties) do not ask the user to authenticate directly instead they consume identity assertions from the identity provider [6]. This could be a good approach to solve the problem of remembering many passwords by reducing to one account at the identity provider, but

does not solve the original problems of passwords because people still could use weak password on the identity provider, and if this gets compromise then all the others websites would be too. Another challenge that this scheme face is the trust in not misuse personal information and widely adoptions by other websites.

Two Factor Authentication. This is possibly one of the most widely used authentication mechanism, which is defined by two of the following three factors; something the user knows (e.g. password), something the user has (e.g. phone, smart cards, tokens) [18], or something the user is (e.g. biometric, like fingerprint or voice print). The big problem with this approach is that the user always needs to carry the second factor device, if they loose or forget the device, they won't be able to login at the website. Users prefers easement and simplicity over security [17]. or if it's intended to use something the user is it will face with the problem of deployability.

Password Managers. Firefox [15], LastPass [12] and PwdHash [16], are some examples of advanced kinds of password managers that are built into browsers: they good to provide different passwords for different sites preventing phishing attacks and password sharing. These advanced password managers, however, have their own set of usability issues [5], like the fact that users no longer know the passwords for certain sites and if they forget their own password for the password manager, it could possibly lost all their information.

3 MFTA Scheme Approach

With this work we propose a variation of multiple factors authentication mechanism, but without the need of a cellphone or hardware token, for that it will use the user's PC features as a second factor for authentication.

Scope. The scheme is aimed at web applications, where the web application provides a service to a third party. This scheme is not general purpose, but targeted an organization that wants to give access only to certain people belonging to a group or sector within an area or region (e.g. Country). Below are outlined the goals for this work.

- Transparent to the user. We are convinced that innovative multifactor approaches are promising, and we believe that the proposed scheme will be an asset if it is transparent to the user, which is why it is proposed as central axis in this work.

- Increased security through second factor mechanism. Second factor authentication mechanisms provides an extra security to web applications, but usually comes at the expense of usability.
- Platform independent. In order to make it useful, the scheme needs to work well on most common browsers, and be operating system independent.
- Contingency mechanism. We provide a contingency mechanism when the user needs for any reason use another computer that it is not his own.

Assumptions. It's assumed that the user only access the web application from a single PC in one place, using this as a second factor device and considering that the PC does not change regularly. Access from any other PC or elsewhere that is not his is considered suspicious. We assume that the user does not change his computer regularly, nor any other feature registered. Considering that this could be implemented in a service company with very sensitive information (e.g. a bank, cooperative, credit card processing, etc), we assume that there is a security administrator that usually monitors the server for alerts. We assume that an attacker somehow obtains the user's credentials via phishing or cross site passwords reuse, and manages to successfully connect to the web application from somewhere unusual. We assume that in some cases users might use weak passwords that are easy to guess. An attacker can perform a man in the middle attack interposing between the user and the server to which user is authenticating, thus obtaining the user's credentials. We allow an attacker introduce some kind of malware (such as Keyloger) or virus and collect information and obtain the password.

4 Architecture

Our scheme use the IP address and the user agent information as additional factors to verify the user authenticity. The scheme architecture is explained below.

When the user enters his username and password into a web application as usual, these are sent to the server along with the IP address and the user agent information. Once we have those information, the MFTA will classify the user agent information as follow: browser, operating system, device (e.g.

personal computer, smartphone, tablet, etc). The IP address will help us determine the network and from where the connection is comming. To analize the user agent, the scheme will use the UADetector [13] library, which it will provide us the information we need. We do this by using a Servlet and the function provided by the API:

```
out.append(agent.getName());
out.append(agent.getOperatingSystem().getName());
```

These features will be saved and compared with the user's PC features the next time the user sign in. As the first time the user will have no data to compare with, the scheme will automatically let the user log in and save the information that correspond to the user's PC. For each user there would be a trusted computer list that is saved on the server, similar to what Google has, so that way it will be possible to compare user's PC features everytime the user sign in.

Contingency Mechanism. Considering that for any reason the user needs to change his computer or some other features that our scheme registered, then it will let the user log in, and send an email to the user and the the security administrator notifying of the change in the users features.

Modes. The proposed scheme could work in two different modes, flexible and strict mode. The first one, could let the user log in from a different computer that never used before, but it will send and alert to his email with the login notification, after that, if the user don't take any action, the computer from where was logged will be included into the registered computer list. The strict mode rejects, and notify the security administrator, any user's login attempt if the user's computer is not in the registered device list, except when an user login for the first time with his computer. A new computer can also be added manually on the registered device list.

5 Evaluation

We evaluate our approach with other two approaches that are the most widely used at the moment, the passwords and Google 2 step verification, considering the usability, deployability and security metrics proposed in Bonneau evaluation framework [4]. We evaluate the flexible and strict mode of the MFTA. The table 1 shows the result of the comparative evaluation of MFTA and some others authentication mechanisms.

	Usability								Deployability						Security									
Scheme	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Propietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trust-Third-Party	Requiring-Explicit-Consent	Unlinkable
Passwords	n	n	y	n	y	y	s	y	y	y	y	y	y	y	n	s	n	n	n	n	n	y	y	y	y
Google 2-SV	n	n	n	n	y	s	s	s	s	n	n	y	y	n	s	s	y	y	n	y	y	y	y	y	y
MFTA flexible	n	n	y	n	y	y	y	y	y	y	s	y	n	y	y	s	s	s	n	s	s	y	y	y	y
MFTA strict	n	n	y	n	y	y	s	y	y	y	n	y	n	y	y	y	y	y	n	y	y	y	y	y	y

Table 1: Comparative evaluation of the MFTA with some others schemes. "y" means the benefit is provided, "s" means the benefit is somewhat provided, while "n" means the benefit is not provided.

Usability. The MFTA flexible mode score better than the Google 2-SV and tight with the password approach. In the *infrequent-errors* the flexible mode offers the benefit because if the user change his computer, it will still be able to log in. However the strict mode won't let the user log in if he change his compute, it will need the intervention of the security administrator to add the manually the new computer to the registered device list, so we considered that the benefit is provided somewhat if the computer change is programmed.

Deployability. Both mode of the MFTA approach are not mature yet, so for the moment they could not provide the benefit of *mature*. We consider the benefit of *server-compatible* is provided somewhat in the flexible mode because no change need to be made to the authentication scheme on the server side, just need to gather some information. On the other hand, the strict mode does not provide that benefit because of the change that needs to be done.

Security. On these metrics we can see that the MFTA, in both modes, provide a better security than password and the strict mode score better over Google 2-SV. We can not say the same about the flexible mode, but considering the scope of the work we believe the MFTA score well on the metrics described in the evaluation framework.

6 Conclusion

In this work we introduced the MFTA approach, which provide a better authentication mechanism for websites considering the scope proposed before. The MFTA provides usability benefit as well as conventional passwords do. At the same time it use some of the user's PC feature to provide a better security like second factor authentication do.

The fact that the user does not need to carry any device with him, provides the ease of use plays a key role in authentication, so that's why we propose the MFTA to be transparent, while increasing the security. We think that this is a viable approach and a base for other possible solutions.

References

[1] Fadi Aloul, Syed Zahidi, and Wassim El-Hajj. Two factor authentication using mobile phones. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pages 641–644. IEEE, 2009.

[2] OpenID Authentication. 2.0-final. *OpenID Foundation website*, 2007.

[3] Joseph Bonneau. Measuring password re-use empirically. *Light Blue Touchpaper*, 2011.

[4] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567. IEEE, 2012.

[5] Sonia Chiasson, Paul C van Oorschot, and Robert Biddle. A usability study and critique of two password managers. In *Usenix Security*, volume 6, 2006.

[6] Alexei Czeskis, Michael Dietz, Tadayoshi Kohno, Dan Wallach, and Dirk Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 404–414. ACM, 2012.

[7] Sabina Datcu. The limits of privacy. is this your password? `http://www.hotforsecurity.com/blog/the-limits-of-privacy-is-this-your-password-865.html`, 2010.

[8] Anthony Elledge. Phishing: An analysis of a growing problem. *Sans Institute*, 2004.

[9] Facebook connect. `https://www.facebook.com/help/405977429438260`, 2008.

[10] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666. ACM, 2007.

[11] Christopher Hadnagy. *Social engineering: The art of human hacking*. John Wiley & Sons, 2010.

[12] Lastpass. `https://lastpass.com/`.

[13] Jaroslav Mallat. Uadetector. `http://user-agent-string.info/`, 2014.

[14] Browserid. `https://login.persona.org/about`.

[15] Firefox password manager. `https://support.mozilla.org/en-US/kb/password-manager-remember-delete-change-passwords`.

[16] Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, and John C Mitchell. Stronger password authentication using browser extensions. In *Usenix security*, pages 17–32. Baltimore, MD, USA, 2005.

[17] Linda Summers. Survey finds nearly half of consumers fail to upgrade software regularly and one quarter of consumers don't know why to update software. http://blogs.skype.com/2012/07/23/intl-tech-upgrade-week/, 2012.

[18] Emilio Valente. Two-factors authentication: Can you choose the right one? *Sans Institute*, 2009.