Push Notification Based Login Using BLE Devices

Gaurav Varshney , Manoj Misra
Department of CSE, Indian Institute of Technology, Roorkee, Uttarakhand, 247667 India
mybbc.dcs2014@iitr.ac.in; manojfec@iitr.ac.in;

Abstract— Due to the consistent number of phishing attacks there is a need to develop secure authentication schemes. These days phishing is mostly carried out using malicious browser extensions or other sophisticated credential stealing techniques such as CR/RT MITM phishing attack. Therefore, it is required to develop secure authentication schemes for handling these attacks. A set of popular authentication schemes has been analyzed in this paper and it is found that they are unsafe against credential stealing attacks such as CR/RT MITM phishing, malicious browser extensions, key loggers and malwares. A novel push notification based scheme that uses BT_{ADDR} as the user identification token and performs real time modification and verification of BLE device (Bluetooth 4.0+ Version) descriptor value for user authentication has been proposed in this paper. The scheme is better in terms of security provided by the existing two factor authentication schemes (OTP/PIN, QR Code/Barcode, Graphical password/ CAPTCHA) and the usual push notification based login authentication schemes that uses username as the user identification token. Scheme has been compared with other popular schemes using the Bonneau et al. assessment framework in terms of usability, deployability and security. The results obtained are promising which suggest Bluetooth based push notification schemes can be a good option for future web authentication.

Keywords—Phishing, RT/CR MITM, Malicious Browser Extensions, Bluetooth, Push notification

I. INTRODUCTION

Phishing attacks [1-7] deceive the user into entering credentials on a phishing website (PW). These credentials are then entered on an authentic website by the phisher either in real time or offline to gain access to user accounts. OTP and PIN based two factor authentication (2FA) schemes [8, 9] are unsafe against active real-time phishing attacks such as RT (Real Time)/CR (Control Relay) MITM (Man In The Middle) phishing attacks [10, 11]. There are other ways also through which attackers can steal user credentials entered on a website. For example, a malicious browser extension (ME)[12-15] installed on user's machine can steal user's credentials by sniffing user key strokes, passwords, credit card information and / or the user desktop screen. A covertly installed spoofed app or browser extension used in the authentication process can also be used to acquire user credentials. In a nutshell if following conditions hold then credentials of any authentication scheme can be accessed by an attacker [16]. The conditions are:

(1) All authentication credentials are entered manually over the website and nothing is stored on a hardware device which is automatically accessed and used during login.
(2) All authentication credentials can be obtained by a PW in real time and can be relayed to the authentic website.

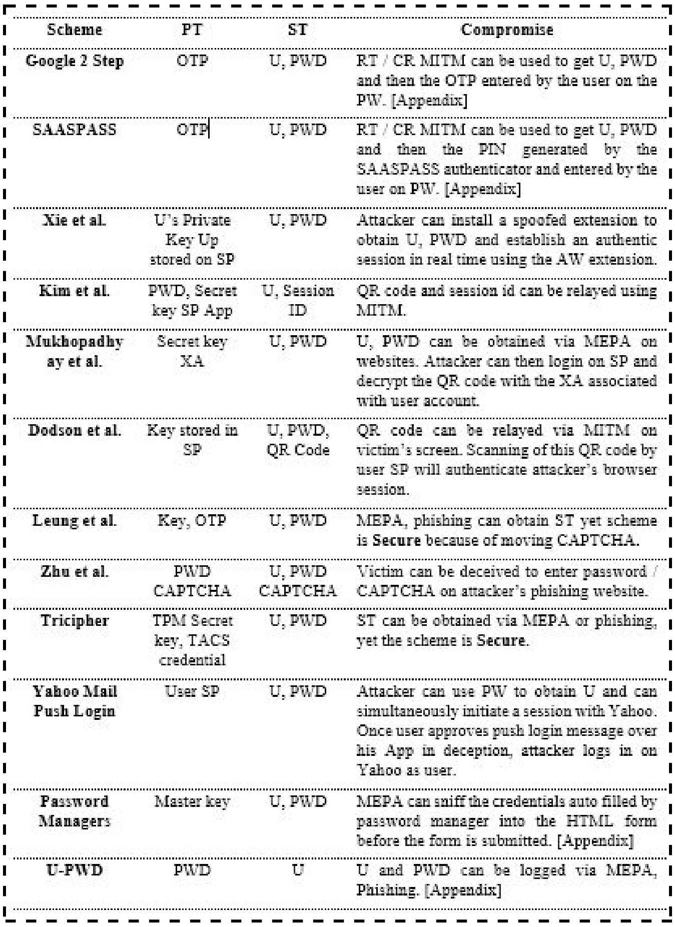
The above conditions hold for most of the popular authentication schemes which are currently used over the web and therefore these schemes are vulnerable to credential stealing attacks of one or the other kind. This paper presents the security analysis of the existing authentication schemes and proposes a variant of the push notification based authentication scheme that uses BLE [17-19]smart devices (Smartphones, Bluetooth beacons, smart bands, health monitors etc. that support Bluetooth 4.0 and above version). The idea is to replace the traditional username used for initiating the push notification by BT_{ADDR} of the registered BLE device. Also, the value of a specific BLE device Descriptor is modified by the website when the user initiates the login. This value is verified by the Smartphone (SP) App when user approves the push notification. This makes it impossible for an attacker to break the authentication (in real-time) even if the BT_{ADDR} of the user's device is known to the attacker.

II. RELATED WORK

The current popular web authentication proposals can be categorized into: *(1) OTP/PIN based 2FA schemes* (Google 2 step [8], SAASPASS [9, 20]) *(2) QR code based schemes* (Xie et al. [21] Kim et al. [10], Mukhopadhyay et al. [22], Dodson et al. [23]) *(3) Graphical password based schemes* (Leung et al. [11], Zhu et al. [24]) *(4) Hardware token based* (Tricipher [16], Yubico [25]), *(5) Push notification based login* (Yahoo [26, 27], SAASPASS [9, 20]) *(6) Password Manager based* (Lastpass [28], Ross et al. [29]). In each of these categories we have considered and analyzed those proposals which are either popularly used or it has been claimed that they are secure against RT/CR MITM attacks. In Figure 2.1 these proposals are compared with respect to the primary (PT) and secondary authentication tokens (ST) used during the authentication process. Figure 2.1 also shows that most of these proposals are either vulnerable to RT/CR MITM phishing or ME based phishing attacks (MEPA).

Most of the current authentication schemes (except the ones which use hardware tokens or complex moving Graphical images, passwords and CAPTCHAS) are insecure against the RT/CR MITM phishing attacks and MEPA. Videos in Appendix show that assumption made by most of the 2FA schemes, that credentials such as OTP/PIN or QR codes cannot be relayed in a limited time frame from a PW to an authentic website (30 seconds assumed by most of the authentication schemes[9]) is not correct and RT MITM attack is possible on these schemes using sophisticated techniques and tools such as selenium. The hardware token based schemes are safe but require user to carry an additional hardware such as security keys which might be inconvenient for most of the users and has an additional cost. Also schemes which use location of user's mobile device for verification during login are vulnerable to location spoofing. Push notification based authentication schemes are new but recent phishing tactics can break these authentication schemes too.

(**Note:** U: Username, PWD: Password)



Scheme	PT	ST	Compromise
Google 2 Step	OTP	U, PWD	RT / CR MITM can be used to get U, PWD and then the OTP entered by the user on the PW. [Appendix]
SAASPASS	OTP	U, PWD	RT / CR MITM can be used to get U, PWD and then the PIN generated by the SAASPASS authenticator and entered by the user on PW. [Appendix]
Xie et al.	U's Private Key Up stored on SP	U, PWD	Attacker can install a spoofed extension to obtain U, PWD and establish an authentic session in real time using the AW extension.
Kim et al.	PWD, Secret key SP App	U, Session ID	QR code and session id can be relayed using MITM.
Mukhopadhyay et al.	Secret key XA	U, PWD	U, PWD can be obtained via MEPA on websites. Attacker can then login on SP and decrypt the QR code with the XA associated with user account.
Dodson et al.	Key stored in SP	U, PWD, QR Code	QR code can be relayed via MITM on victim's screen. Scanning of this QR code by user SP will authenticate attacker's browser session.
Leung et al.	Key, OTP	U, PWD	MEPA, phishing can obtain ST yet scheme is **Secure** because of moving CAPTCHA.
Zhu et al.	PWD CAPTCHA	U, PWD CAPTCHA	Victim can be deceived to enter password / CAPTCHA on attacker's phishing website.
Tricipher	TPM Secret key, TACS credential	U, PWD	ST can be obtained via MEPA or phishing, yet the scheme is **Secure**.
Yahoo Mail Push Login	User SP	U, PWD	Attacker can use PW to obtain U and can simultaneously initiate a session with Yahoo. Once user approves push login message over his App in deception, attacker logs in on Yahoo as user.
Password Managers	Master key	U, PWD	MEPA can sniff the credentials auto filled by password manager into the HTML form before the form is submitted. [Appendix]
U-PWD	PWD	U	U and PWD can be logged via MEPA, Phishing. [Appendix]

Figure 2.1. Authentication proposals and their security

Consider the example of Yahoo mail push notification based login. In this scheme user enters the username on the Yahoo mail login page on his web browser. After receiving the username, server sends the push notification to the Yahoo App installed on user's SP. Once user approves the push notification received on the Yahoo App installed on his SP, he gets the access to his Yahoo account on the web browser. Though the scheme looks safe but a phisher can steal access to the user account by following these steps.

(1) A ME installed by the phisher can steal the username entered by the user on authentic Yahoo login page (even before user clicks the submit button) in real time.
(2) ME updates the submit action link of Yahoo web page loaded on the client browser with the link of a PW. Hence user is redirected to a phishing page which looks like a Yahoo login page waiting for user approval of push notification.
(3) In the mean-time attacker enters the acquired username on the authentic website opened on his browser.
(4) Yahoo after receiving the correct username from the attacker's browser session generates a push notification on the user's SP App.
(5) The user approves the push notification received on his SP assuming that the push notification is for his browser session with the Yahoo.
(6) This gives the attacker access to the user's account. User can then be redirected to the authentic Yahoo login page or an error page can be displayed on the PW.

The other way of deceiving user is directly through a PW.

(1) Phisher can send a phished Yahoo login page link to user via an email or other medium.
(2) User will open the phished Yahoo login page and enter his username. The username will be received and entered by the attacker on the real Yahoo login page opened on his browser. On the PW, a web page will be displayed waiting for user approval of push notification.
(3) Yahoo will send a push notification to the user SP App.
(4) User will approve the push notification message thinking that push notification is for his browser session with the Yahoo.
(5) This will allow the attacker to login into the user account.

To improve the security of the existing push notification based login authentication, we propose a push notification based login authentication scheme that uses BLE devices. Instead of username the BT_{ADDR} is used as the user identification token because it can be acquired automatically by the websites. This makes it impossible for the attackers to obtain user identification token via host based key logging attacks and sniffing attacks done via MEs. Additional security is provided by modifying the characteristic descriptor value of the BLE device. The value is modified by the website and communicated to the user SP App in push notification message. After SP App verifies the modified value and user approves the push notification, website allows the user to login. The next section discusses the proposed authentication scheme

III. Bluetooth Based Authentication over Websites (BBAW)

A. Assumptions

It is assumed that the user logs in on the website using a web browser installed on his desktop/laptop or through a App installed on his SP. Laptop or desktop has a Bluetooth adapter. User either uses his SP (the one on which website App is installed) or a separate BLE device (smart wearable, health monitor, BLE beacons, SP) as the registered BT device during web registration and login. The Attacker is capable of: (1) installing malware on user desktop / laptop, (2) logging user keystrokes / screen with the help of a ME, and / or (3) sniffing and spoofing of user BT_{ADDR} on his SP or BLE device.

B. Abbreviations

HTTPs (A, B, C…)	HTTPs message with parameters A, B, C and so on.
SEND_OTP(PN)	Sends OTP to the user registered phone number (PN)
V_{DB} (Parameter)	Verifies the parameter value with the corresponding value in the database

REG_{DB} (Parameters)	Registers the user and creates a new tuple entry in the database with parameters
Select (S, C, D, Val)	Server selects a combination of Service (S), Characteristic (C), Descriptor (D) and its value (Val) to be modified in the device
Write_BLE (S, C, D, Val)	Writes value (Val) of the Descriptor (D) belonging to specific Service (S) and Characteristic (C) in the BLE device
Send Notification (App)	Sends a push notification message to the user SP App
Match	A binary variable. 0 if descriptor value is matched else 1
Login Attempt Notify (Parameters)	Delivers push notification to the SP App with parameters
Response (A, Match)	Sends user response of push notification (A: Approval or Rejection) and value stored in variable Match to the server
Server_Verify ()	Allows user to login into his account if user approves the push notification and the value of Match is 0.

C. Proposed Authentication Scheme using BT/BLE Devices

The proposed scheme for includes two phases: (1) Web Registration Phase (WRP) and (2) Web Login Phase (WLP).

WRP: In WRP a new user registers himself on the website. During registration phase the user must have a BLE device connected to his desktop.

(1) In WRP user opens the website registration page and enters his personal information including the Name (N), Email Id (EM), Password (PWD), Phone Number (PN) on the website registration page.

(2) Once the user clicks on the submit button after entering his personal details the website uses the Web Bluetooth API to search the nearby BLE devices (ready to be paired) to the user desktop/laptop and displays their names to the user on a pop up window.

(3) The user is asked to select a device from the set of BLE devices displayed on the pop up window. The BT_{ADDR} of the selected device and the personal information entered by the user in step (1) is sent to the website over the HTTPs channel.

(4) The website, after receiving the information, sends an OTP (SEND_OTP) to the registered mobile number (PN) of the user for verification.

(5) Once the correct OTP is entered by the user and verified by the server (V_{DB}) user is successfully registered with the website and a database entry is created (REG_{DB}). The messages exchanged during WRP are shown in Fig 3.1.

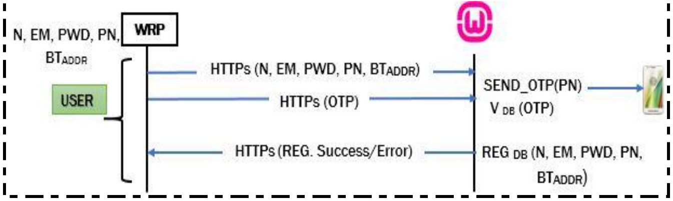

N, EM, PWD, PN, BTADDR
WRP
USER
HTTPs (N, EM, PWD, PN, BTADDR)
HTTPs (OTP)
HTTPs (REG. Success/Error)
SEND_OTP(PN)
V DB (OTP)
REG DB (N, EM, PWD, PN, BTADDR)

Figure 3.1. BBAW: WRP Phase

WLP: To login into his website account using a desktop/laptop user must be logged in to his website account on the website's SP App. Logging over the SP App is needed for any push notification based schemes to work as the push notification is sent to the SP App on which user is logged in.

The user can install the App of the website on his SP. After the App installation, he should connect his SP with his registered BLE device. In case his SP is the registered BLE device there is no need to connect any other device with the SP. After this, user must enter his EM and PWD over the App login screen. Once the user presses the submit button he is required to select one of the BLE device from the set of all connected BLE devices shown on a pop up window. The BT_{ADDR} of the selected device is sent to the server along with the user EM and PWD for verification. Once verified user gets logged in to his account on the SP App. As most of the users do not log out from their accounts on SP Apps the re-login over SP App happens infrequently.

Now user must follow steps given below to login into his website account using the desktop:

(1) The user must open the website login page (Open (Login URL)) on the browser.

(2) On the login page user, should click on "Select BLE Device" button. It is assumed that user has already connected his registered BLE device with the desktop/laptop.

(3) A pop up window is displayed on the website login page with the names of the nearby BLE devices fetched by the website. User must select and pair his registered BLE device from these devices.

(4) Once the user selects the BLE device the website fetches the BT_{ADDR} of the device using Web Bluetooth API[30]and modifies a specific BLE characteristic descriptor [31]value of the device (Write_BLE (S, C, D, Val)).

(5) The website chooses a combination of GATT[32] Service (S), Characteristics (C) and a specific Descriptor (D) value for modification. For example, the website can choose battery_service (S) and battery_level (C) and can change its characteristic_user_description (D) value (Val) to 100 using Web Bluetooth write descriptor value method.

(6) After changing the specific descriptor value the website sends the BT_{ADDR} of the device to the webserver.

(7) The website server first identifies the user associated with the BT_{ADDR} received and then generates a push notification message (Login Attempt Notify) to the corresponding SP App on which the user is currently logged in.

(8) The push notification message additionally contains the BT_{ADDR} of the BLE registered device, the selected Service (S), Characteristic (C), Descriptor (D) and the modified Value (Val).

(9) The SP App matches the BT_{ADDR} with its own BT_{ADDR} and the BT_{ADDR} of the paired devices. The Descriptor (D) Value (Val) of the matched device is compared with the value received in the push notification. If both are same the Match variable is set to 0 else, it remains 1 (Default).

(10) Once the user approves the push notification message the SP App sends the user response (Approval/Rejection) and the value of Match variable to the server.

(11) After the server receives the user approval it allows user to login to his account if the value of Match variable is 0. The message exchanges (considering a separate BLE device used during WRP) are shown in Figure 3.2.

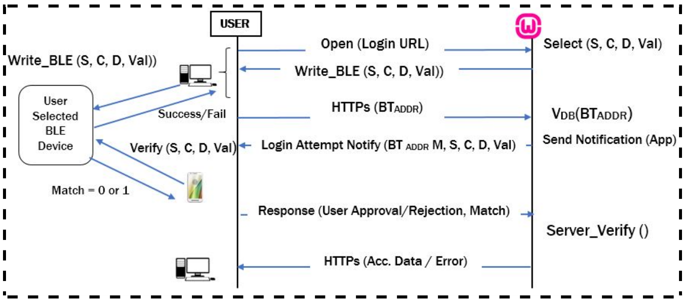


Figure 3.2. BBAW: WLP Phase

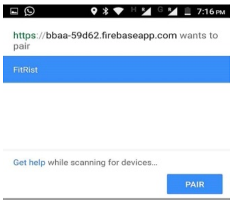
It should also be noted that the original value of the Descriptor in the BLE device (before the login is initiated) is also fetched by the website and communicated to the SP App in the push notification message. This original value of the Descriptor is set in the BLE device by the SP App after the modified Descriptor value gets is verified. One should also note that any BLE device can be used during the registration and hence the user is free to choose any BLE device (Example: smart wearable, health monitor, smart watch, smartphone, BLE beacon etc.) based on his/her comfort and device availability.

IV. Implementation and Test Setup

To test the website registration phase and login phase we opened the website on the Chrome browser installed on a SP running Android Marshmallow operating system. The SP was used in place of desktop because the Web Bluetooth API[17, 30, 31] used for fetching the BT_{ADDR} of a BLE device and for writing BLE descriptor value by a website has been released currently for Android OS, Chrome OS and few versions of Linux (support for the API on Chrome and other browsers over Windows platform is to be released in a few days). The browser development companies (such as Firefox, IE and Google Chrome) and World Wide Web (WWW) community is committed for the release of the Web Bluetooth API on all platforms. Hence though the proposed scheme can currently be used on Chrome OS, Android OS, Linux but after some time the developers can use the proposed scheme on all platforms[33]. We used a SP (Motorola X Play Handset, having Android Marshmallow OS installed on it) as a desktop/laptop and a BLE device (Intex FitRist Smart wearable shown in Figure 4.1) as the registered BLE device for user registration and login.

Figure 4.1. BLE Device (Intex FitRist)

The App was installed on another SP (Motorola E 3 Handset). The website registration and login page was hosted on Heroku server. The website was written in HTML and JS and Web Bluetooth APIs code was incorporated in the JS. The BT_{ADDR} of the BLE device was fetched by the website using Web Bluetooth APIs during WRP and WLP phase. Appendix shows a video made during our testing. Figure 4.2 shows the pop up window where the connected BLE devices (FitRist) are displayed. Push notification received by the user's SP App is shown in Figure 4.3



https://bbaa-59d62.firebaseapp.com wants to pair

FitRist

Get help while scanning for devices...

Figure 4.2 Device Selection Popup

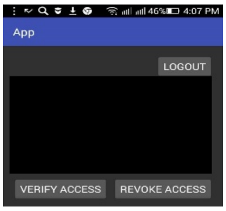


Figure 4.3 Push Notification

As the area of IOT devices and APIs to access the data from smart wearable via Bluetooth is emerging at a fast speed we expect to receive stable Web Bluetooth APIs for Linux and Windows platform very soon[33, 34]. Demo implementations of the scheme on Linux and Windows platform will be released after these APIs are available.

V. Security Analysis

A. *Security against RT MITM phishing*

Attacker must do the following for RT MITM phishing attack:

(1) Sniff and spoof the user BT_{ADDR} on his BLE device.
(2) Redirect the user to the PW.
(3) Open the authentic website on the attacker browser.
(4) Authentic website sniffs the BT_{ADDR} of the attacker's device and modifies a specific BLE descriptor with a random value.
(5) The attacker sniffs the BLE descriptor value based on the commands fired by the website using loggers in the operating system or he can directly check the specific Descriptor and its changed value in his BLE device.
(6) Attacker uploads a file containing the Descriptor details and its value to be read by the dynamic script running over the PW.
(7) The PW reads the newly uploaded details of the Descriptor and its value and uses the Web Bluetooth API to write the BLE device descriptor of the user BLE device.

The attack will succeed if the time taken by the attacker to update the BLE Descriptor value in user's device via the PW i.e. *T(Attacker)* is lesser than the time it takes for the authentic website to send push notification to the SP App and the SP App to verify the user's BLE device descriptor value *T(SmartphoneApp)*.

T(Attacker)= T(SniffBLEChange) + T(UploadChangeFile) + T(Write_BLE_PW)

T(SmartphoneApp)= T(Comm_UID) + T(User_Verification) + T(Comm_Push_Login_App) + T(SmartphoneBLEVerification)

*Where, **T(Attacker):** Total time it takes for the attacker to retrieve BLE Descriptor change from his BLE Device and Write it through PW on user's BLE device. **T(SniffBLEChange):** Time to sniff BLE Descriptor Change on attacker's BLE device via PW. **T(UploadChangeFile):** Time needed for uploading the file with the modified descriptor value over web server. **T(Write_BLE_PW):** Time needed by the Web Bluetooth API to write BLE Descriptor value on user BLE device via PW. **T(SmartphoneApp):** Time needed by the scheme to receive user information via authentic website and verify the BLE descriptor information by the Smartphone App. **T(Comm_UID):** Time, it takes for the website to communicate the user identification token (BT_{ADDR}). **T(User_Verification):** Time needed by the* *server for user verification of UID.* ***T(Comm_Push_Login_App):*** *Time needed for communicating the Push notification to the Smartphone App via Server***.** ***T(SmartphoneBLEVerification):*** *Time needed by the Smartphone App to match the BLE descriptor in the user BLE device*

We found that *T(UploadChangeFile)* is the most time consuming operation during the process. If the best-case time of *T(UploadChangeFile)*is greater than worst case time of *T(SmartphoneApp)* the attack can never happen. To study the attack possibility scenario, we did 100 experiments. We found that in the worst-case *T(SmartphoneApp)* is 2.5 seconds. While the best-case time of *T(UploadChangeFile)* was 3.5 seconds for uploading a Yahoo Login HTML page of 100 KB. To cross check our findings we also performed experimentations on the Yahoo push notification scheme and found that the time *T(SmartphoneApp)* – *T(SmartphoneBLEVerification)* was about 2 seconds. Hence *T(SmartphoneApp)* will always be less than *T(Attacker)*. This concludes that during a RT MITM phishing attack the SP App will verify the old value of the BLE device Descriptor. Hence RT MITM phishing attack is not possible.

B. CR MITM phishing attack

CR MITM phishing is not possible because the use of BT$_{ADDR}$ as the user identification token. The username entered by the user can be obtained by the attacker on his browsing session by relaying his browser screen on user desktop but it is impossible to access the BT$_{ADDR}$ of the device connected to the user PC via Web Bluetooth APIs as Web Bluetooth APIs can only access the BLE devices physically connected to the attacker's PC.

C. ME based credential stealing attacks

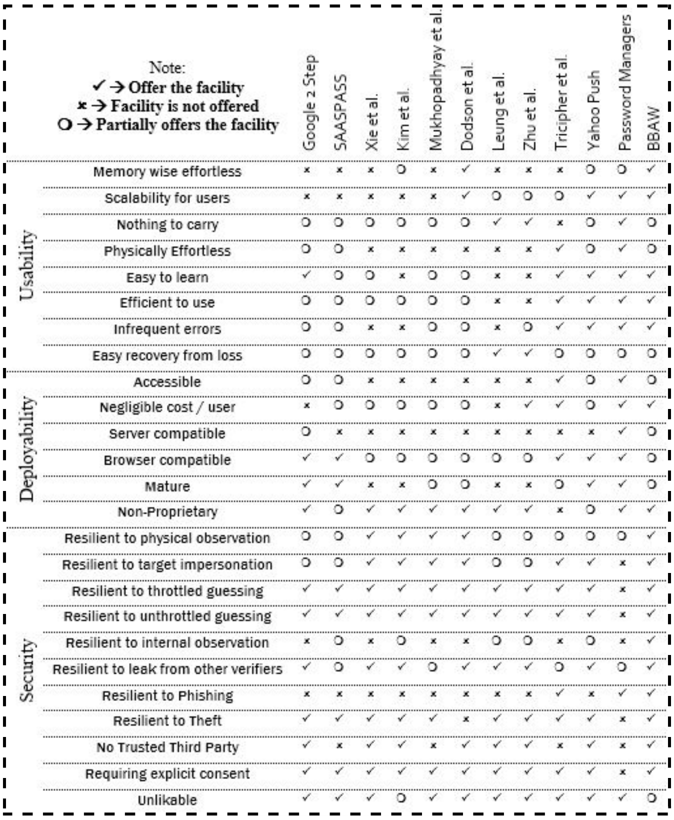
As user, does not type any information on the websites the credentials stealing attacks which happens through keystroke logging, PWD and HTML form data sniffing cannot happen.

D. Host malware based keystroke logging

Keystroke logging based attacks that steal credentials entered by the user over websites can also be avoided with the use of BT$_{ADDR}$ as the user identification token as BT$_{ADDR}$ is automatically retrieved by the websites. Malwares on host which have access to browser memory or cookies cannot harm the scheme as there is no information stored on the desktop.

VI. Assessment with Bonneau et al. Framework

We used the popular Bonneau et al [35] authentication scheme assessment framework to compare our scheme with other schemes. The assessment framework compares the authentication schemes based on their usability, deployability and security. We have taken most of the values from [35]. Other values assigned to schemes in the Figure 6.1 are based on the earlier discussion in Section 2 and from Figure 2.1. We found that our proposed scheme offers most of the benefits of the Bonneau et al. framework completely and some of the benefits are offered partially. For example, it requires the user to carry the SP which comes under *Quasi nothing to carry* and hence the nothing to carry benefit is offered partially. Also, it is not *physically effortless* because it still needs some explicit physical efforts such as pairing the BLE device with the desktop. There may be some efforts to recover the account when your BLE registered device is stolen as with any hardware token based authentication scheme. It is not completely *Accessible* to all as it may need some explicit knowledge of pairing of BLE devices. It is very basic still some users might need some knowledge or training to use or access it. It needs some modifications on the server side and on the client side to support the BLE device access and hence partially offers the *Server compatible and Browser compatible* benefits. The scheme also is not completely *Mature* as it has been implemented and proposed and not in rigorous public use. It also does not provide the benefit of being *Unlinkable* as user login on different websites is linked with the BT$_{ADDR}$ and hence a user can be tracked.



	Note: ✓ → Offer the facility × → Facility is not offered O → Partially offers the facility	Google 2 Step	SAASPASS	Xie et al.	Kim et al.	Mukhopadhyay et al.	Dodson et al.	Leung et al.	Zhu et al.	Tricipher et al.	Yahoo Push	Password Managers	BBAW
Usability	Memory wise effortless	×	×	×	O	×	✓	×	×	×	O	O	✓
	Scalability for users	×	×	×	×	×	✓	O	O	O	✓	✓	✓
	Nothing to carry	O	O	O	O	O	O	✓	✓	×	O	✓	O
	Physically Effortless	O	O	×	×	×	×	×	×	✓	O	✓	O
	Easy to learn	✓	O	O	×	O	O	×	×	✓	✓	✓	✓
	Efficient to use	O	O	O	O	O	O	×	×	✓	✓	✓	✓
	Infrequent errors	O	O	×	×	O	O	×	O	✓	✓	✓	✓
	Easy recovery from loss	O	O	O	O	O	O	✓	✓	O	O	O	O
Deployability	Accessible	O	O	×	×	×	×	×	×	✓	O	✓	O
	Negligible cost / user	×	O	O	O	O	O	×	✓	✓	O	✓	✓
	Server compatible	O	×	×	×	×	×	×	×	×	×	✓	O
	Browser compatible	✓	✓	O	O	O	O	O	O	✓	✓	✓	O
	Mature	✓	✓	×	×	O	O	×	×	O	✓	✓	O
	Non-Proprietary	✓	O	✓	✓	✓	✓	✓	✓	×	O	✓	✓
Security	Resilient to physical observation	O	O	✓	✓	✓	✓	O	O	O	O	O	✓
	Resilient to target impersonation	O	O	✓	✓	✓	✓	O	O	✓	✓	×	✓
	Resilient to throttled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
	Resilient to unthrottled guessing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
	Resilient to internal observation	×	O	×	O	×	×	O	O	×	O	×	✓
	Resilient to leak from other verifiers	✓	O	✓	✓	O	✓	✓	✓	O	✓	O	✓
	Resilient to Phishing	×	×	×	×	×	×	×	×	✓	×	✓	✓
	Resilient to Theft	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	✓
	No Trusted Third Party	✓	×	✓	✓	×	✓	✓	✓	×	✓	×	✓
	Requiring explicit consent	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
	Unlikable	✓	✓	✓	O	✓	✓	✓	✓	✓	✓	✓	O

Figure 6.1. Comparison with other schemes using Bonneau et al.

VII. Limitations

No proposal can be perfect and when there are benefits there are some limitations or drawbacks too. One of the limitation of the proposed scheme is that the users must possess a BLE device /Smartphone during registration and login. It is identified from reports that 80 percent of the Internet users use a Smartphone [36]. This means that 20 percent of the users may not be able to use it because they might not have a Smartphone for installing the Smartphone App. Others who have a Smartphone may need an initial training about pairing of Bluetooth devices etc. if they have not utilized Bluetooth technology in the past. These requirements may limit the number of users. However we assume that the users who are looking for high level of security will have the knowledge of the Bluetooth technology and will possess a Smartphone or buy one for better security during authentication.

VIII. Conclusions and Future Work

A push notification based authentication scheme that use BLE devices is proposed in this paper. BT_{ADDR} of the BLE devices is used for user identification during the login. Also, the value of a specific BLE device Descriptor is modified by the website when the user initiates the login. This value is verified by the SP App when user approves the push notification. This concept of real-time modification and verification of BLE descriptor value makes it impossible for an attacker to break the scheme (in real-time) even if the BT_{ADDR} of the user's BLE device. Analysis shows that the scheme is secure against RT/CR MITM phishing, malicious extension based phishing attacks and attacks that arise from key loggers and malwares on host. Bonneau et al. framework of security assessment provides indication that the scheme is usable, easily deployable and secure. In future, we look for more research and development, usability pilot surveys and launching of commercial implementations of the proposal on websites. Also in future researchers can find scope for possibilities in using the proposed scheme to improve other schemes such as the schemes used for ownership verification [37].

APPENDIX

- Video demonstrate login using the proposed scheme. https://youtu.be/C1y18B3ayEk
- A video created to demonstrate hacking credentials auto filled by password managers (LASTPASS) using MEPA can be accessed here: https://youtu.be/uNwahzlzpSg
- A video created to demonstrate the MITM phishing attack for hacking username and password of a Gmail account can be accessed here:https://youtu.be/nccDJMIkJtc
- A video breaking SASSPASS authentication using RT MITM phishing can be accessed here: https://youtu.be/y-w0RfCaBrQ
- A video demonstrating hacking user accounts through relay of QR codes used in WhatsApp web authentication can be accessed here: https://youtu.be/6FThk1Iystw

REFERENCES

[1] G. Varshney, M. Misra, and P. K. Atrey, "A phish detector using lightweight search features," *Computers & Security,* vol. 62, pp. 213-228, 2016.

[2] G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," *Security and Communication Networks,* vol. 9, pp. 6266-6284, 26 OCT 2016 2016.

[3] N. Abdelhamid, "Multi-label rules for phishing classification," *Applied Computing and Informatics,* vol. 11, pp. 29-46.

[4] R. C. Dodge Jr, C. Carver, and A. J. Ferguson, "Phishing for user security awareness," *Computers & Security,* vol. 26, pp. 73-80, 2007.

[5] C. Ee Hung, C. Kang Leng, S. San Nah, and T. Wei King,"Phishing Detection via Identification of Website Identity," in *IT Convergence and Security (ICITCS), 2013 International Conference on*, 2013,pp.1-4.

[6] M. He, S.-J. Horng, P. Fan, M. K. Khan, R.-S. Run, J.-L. Lai, *et al.*, "An efficient phishing webpage detector," *Expert Systems with Applications,* vol. 38, pp. 12018-12027, 2011.

[7] J. Hong, "The state of phishing attacks," *Commun. ACM,* vol. 55, pp. 74-81, 2012.

[8] Google. (2015). *Stronger security for your google account*. Available: https://www.google.com/landing/2step

[9] I. Barker. (2015). *Saaspass makes two-factor authentication available to the masses*. Available: https://betanews.com/2015/01/15/saaspass-makes-two-factor-authentication-available-to-the-masses/,

[10] S.-H. Kim, D. Choi, S.-H. Jin, and S.-H. Lee, "Geo-location based QR-Code authentication scheme to defeat active real-time phishing attack," in *Proceedings of the 2013 ACM workshop on Digital identity management*, 2013, pp. 51-62.

[11] C.-M. Leung, "Depress phishing by CAPTCHA with OTP," in *Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on*, 2009, pp. 187-192.

[12] M. M. P. Center. (2013). *Browser extension hijacks Facebook profiles*. Available: https://blogs.technet.microsoft.com/mmpc/2013/05/10/browser-extension-hijacks-facebook-profiles/

[13] C. Hoffman. (2017). *Beginner geek: Everything you need to know about browser extensions*. Available: https://www.howtogeek.com/169080/beginner-geek-everything-you-need-to-know-about-browser-extensions/

[14] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson, "Hulk: Eliciting Malicious Behavior in Browser Extensions," in *USENIX Security*, 2014, pp. 641-654.

[15] N. Utakrit, "Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers," presented at the 7th Australian Information Security Management Conference, Perth, Western Australia, 2009.

[16] TRICIPHER. (2016). *Preventing man in the middle phishing attacks with multi-factor authentication*. Available: https://www.globaltrust.it/documents/press/phishing/PhishingSolutionWhitepaper.pdf,

[17] F. Beaufort. (2016). *Interact with Bluetooth devices on the Web*.

[18] A. Developers. (2017). *Bluetooth Low Energy*. Available: https://developer.android.com/guide/topics/connectivity/bluetooth-le.html

[19] F. Martelli, "BluetoothÂ® low energy," *University of Bologna,* vol. 25, 2014.

[20] SAASPASS. (2017). *Two-factor authentication with proximity uses ibeacon bluetooth low energy (ble) to authenticate users instantly*. Available: https://saaspass.com/technologies/proximity-instant-login-two-factor-authentication-beacon.html

[21] M. Xie, Y. Li, K. Yoshigoe, R. Seker, and J. Bian, "CamAuth: Securing Web Authentication with Camera," in *High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on*, 2015, pp. 232-239.

[22] S. Mukhopadhyay and D. Argles, "An Anti-Phishing mechanism for single sign-on based on QR-code," in *Information Society (i-Society), 2011 International Conference on*, 2011, pp. 505-508.

[23] B. Dodson, D. Sengupta, D. Boneh, and M. S. Lam, "Secure, consumer-friendly web authentication and payments with a phone," in *International Conference on Mobile Computing, Applications, and Services*, 2010, pp. 17-38.

[24] B. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "Captcha as Graphical Passwords-A New Security Primitive Based on Hard AI Problems," *IEEE transactions on information forensics and security,* vol. 9, pp. 891-904, 2014.

[25] Yubico. (2017). *Fido u2f (universal 2nd factor)*. Available: https://www.yubico.com/about/background/fido/

[26] Yahoo. (2016). *Yahoo Sign In*. Available: https://login.yahoo.com/

[27] Urbanairship. (2017). *Push Notifications Explained*. Available: https://www.urbanairship.com/push-notifications-explained

[28] Lastpass. (2017). *Lastpass*. Available: https://www.lastpass.com/

[29] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger Password Authentication Using Browser Extensions," in *Usenix security*, 2005, pp. 17-32.

[30] W. C. C. Group. (2017). *Web Bluetooth*. Available: https://webbluetoothcg.github.io/web-bluetooth/

[31] Google. (2017). *Web Bluetooth / Write Descriptor Sample*. Available: https://googlechrome.github.io/samples/web-bluetooth/write-descriptor.html

[32] Bluetooth. (2017). *GATT Overview*. Available: https://www.bluetooth.com/specifications/gatt/generic-attributes-overview

[33] GITHUB. (2017). *Implementation Status: Web Bluetooth*.

[34] U. Shaked. (2016). *Is Now a Good Time to Start using Web Bluetooth? (Hint: Yes, yes it is.)*. Available: https://medium.com/@urish/is-now-a-good-time-to-start-using-web-bluetooth-hint-yes-yes-it-is-99e998d7b9f6

[35] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Security and Privacy (SP), 2012 IEEE Symposium on*, 2012, pp. 553-567.

[36] R. Sukhraj. (2016). *31 Mobile Marketing Statistics to Help You Plan for 2017*. Available: https://www.impactbnd.com/blog/mobile-marketing-statistics-for-2016

[37] S. Shivani, P. Singh, and S. Agarwal, "A Dual Watermarking Scheme for Ownership Verification and Pixel Level Authentication," in *Proceedings of the 9th International Conference on Computer and Automation Engineering*, pp. 131-135.