A New Secure Authentication Scheme for Web Login Using BLE Smart Devices

Gaurav Varshney [1], Manoj Misra [1], Pradeep Atrey [2]
[1] Department of CSE, Indian Institute of Technology, Roorkee, Uttarakhand, 247667 India
[2] Computer Science Department, State University of New York at Albany, NY, USA
mybbc.dcs2014@iitr.ac.in; manojfec@iitr.ac.in; patrey@albany.edu

***Abstract*—Existing user authentication schemes used for login to a website are incapable of handling recent phishing attacks such as real time (RT) / control relay (CR) man in the middle (MITM) attack and attacks launched via covertly installed malicious browser extensions (MEs). Two factor authentication schemes such as Google 2 Step verification, SAASPASS, QR code, graphical password and push notification based login schemes can be compromised using RT / CR MITM phishing attacks. Hardware token based schemes are safe but the extra cost of the hardware token makes them unattractive to users. Therefore, there is a need to develop new authentication schemes which are hard for an attacker to compromise but easy for users to understand and utilize. This paper analyzes existing authentication schemes to identify the research gaps and then proposes a secure authentication scheme which uses Bluetooth Low Energy (BLE, BT 4.0+ version) devices for user identification and which can handle RT/CR MITM phishing attacks, attacks launched via malicious browser extensions and app spoofing via attackers. The proposed scheme is location/client system independent and is secure from Bluetooth address spoofing attacks.**

Keywords- phishing; BLE; Bluetooth; malicious browser extension; login; authentication.

I. Introduction

Phishing is an easy way to hack user credentials by deception. In the past, attackers have compromised many one way and single factor user authentication schemes by creating phishing websites similar in appearance to legitimate websites and then spreading these websites via emails, DNS poisoning, Tabnabbing, content injection or Browshing, etc.[1]. Attackers are now using RT/CR MITM [2-4] phishing attacks and MEs which can perform screen logging, key stroke logging or password sniffing to hack user's credentials (see in Appendix). Covertly installed MEs via malicious insiders [5-7] and remote desktop capturing modules are helping attackers to hack user credentials even in case of multifactor authentication schemes.

We analyzed Google 2 step verification [8] that uses OTP and the PIN based authentication of the SAASPASS [9] authenticator app. We successfully implemented and verified RT MITM phishing scenarios to break secret PIN based SAASPASS 2FA and OTP based Google 2 step 2FA. The video in *Appendix* shows the complete procedure. We tested recent QR code and Barcode based multifactor authentication schemes [2, 3, 10, 11] for latest phishing attacks. We found that all of these techniques are vulnerable to RT/CR MITM phishing attacks and attacks launched via MEs. To prove our claim, we successfully performed a phishing attack on WhatsApp web, which requests the user to scan a QR Code for login. For details please see the video in Appendix. The graphical password based schemes such as [4] are vulnerable to CR MITM attack. These schemes are also user unfriendly. The approach in [12] can fail if the ME collects the screen logs. These logs can be used by the attacker to identify exact CAPTCHA characters entered by the user. It is very hard to phish or hack a user's account when they use security keys or hardware tokens such as Yubikey [13]. However, they are not widely used as they require the user to carry an extra device and there is a cost involved in purchasing these devices. Tricipher [14] also proposed a multipart credential based scheme that uses a TPM module at the client end and a hardware security key. Push notification based login based authentication schemes [15, 16] are vulnerable to RT/CR MITM attacks. An attacker can compromise them by sending a phishing website link to user. The user under deception will enter his user name on the phishing website. Attacker after receiving the user name will enter it in real time on the authentic website. The push notification corresponding to the user account will be sent to the user's Smartphone app. User will approve the notification for login under impression that the push notification message is to authenticate his browser session. Password managers such as LASTPASS are vulnerable to ME based phishing attacks. For details please see the video in the Appendix.

A. Recommendations

Based on the above study we identified that an authentication scheme must have the following design features to handle sophisticated credential stealing attacks launched via malicious insiders/outside attackers through covertly installed ME/apps and via remote desktop relay and monitoring modules to cause RT/CR MITM phishing attacks.

- In any secure authentication scheme, the user must enter a minimal number of authentication credentials manually, and at least one of the credentials must be acquired

978-1-5386-0533-2/17/$31.00 ©2017 IEEE

automatically in a way which makes it difficult for the attacker to acquire and relay. This is because if all the credentials are entered manually by the user, it becomes easy for the attacker to acquire them with the help of a phishing website.

- The user identification token such as UNAME should be replaced by a token that is hard to acquire and even if the attacker acquires it, it must be difficult for the attacker to use or relay it.
- Websites must not receive user credentials through key strokes as the MEs can log key strokes and can perform password sniffing.

To address the vulnerabilities of existing schemes, we propose an authentication scheme which uses the Bluetooth hardware address (BTADDR) of BLE (BT 4.0+ version) smart devices (a Smartphone, a BT enabled beacon, a smart wearable, a health monitor etc.) as a user identification token instead of the traditional UNAME. User enters his password on a browser extension and the use of app id as a secret key for encryption makes the scheme secure from sophisticated phishing attacks or attacks via MEs. The proposed Bluetooth Based Authentication scheme using browser Extensions (BBAE) is discussed in the next section.

II. Bluetooth Based Authentication via Browser Extension (BBAE)

BBAE includes: (1) Web Registration Phase (WRP), (2) Extension Registration Phase (ERP), and (3) Login Phase (LP).

A. Web Registration Phase (WRP)

User (U) enters his Name (N), PWD, Email ID (EM) and Phone Number (PN) on the registration page of the website. The U also selects a BLE smart device connected to his PC (SP or a smart wearable) that he wants to register with the website. Once the user selects the BT device, the BT$_{ADDR}$ is fetched by the Web Bluetooth API of the website. Then, information entered by the user along with the BT$_{ADDR}$ are sent to the website. The website verifies the user by sending an OTP (SEND_OTP) to the mobile phone of the user. Once the correct OTP is returned by the user and is verified (V$_{DB}$) by the website, the website generates (GENERATE) a 16-character SALT and associates it with the user account. SALT is generated using AES in CTR mode with count value set to 0 and the current timestamp of the server used as IV.

Also, a unique 16-character string generated using a pseudo random number generator is used as the input secret key. The 16-character string of the yield of one CTR block of operation is used as the SALT. This SALT is used to encrypt (E) the app instance id (AID$_{CX}$) at a later stage. It is assumed that there is no attack during WRP and the system is safe from host based key loggers. Figure 1 shows the message exchanges.

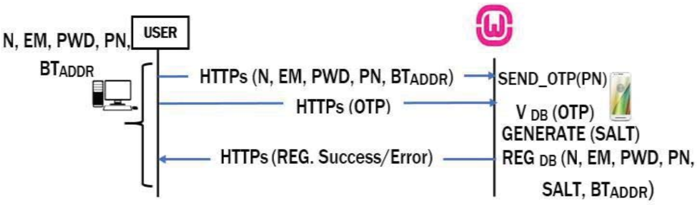

N, EM, PWD, PN,
BTADDR
USER
HTTPs (N, EM, PWD, PN, BTADDR)
HTTPs (OTP)
HTTPs (REG. Success/Error)
SEND_OTP(PN)
V DB (OTP)
GENERATE (SALT)
REG DB (N, EM, PWD, PN, SALT, BTADDR)

Figure 1. Web Registration Phase

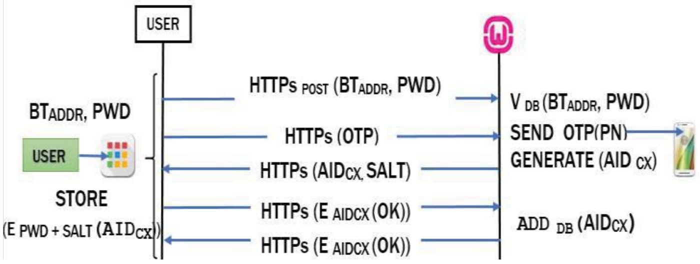


Figure 2. Extension Registration Phase

B. Extension Registration Phase (ERP: one time)

The legitimate website provides a Chrome Extension (CX) that can be downloaded from the browser extension web stores. During installation, when the user registers the CX with the website, the registered BT device must be connected to the PC. The user enters his PWD on the CX registration page, and the BT$_{ADDR}$ of the connected device and the PWD are verified by the website. An OTP is sent to the user's registered PN. After the OTP is verified, the server generates an app instance id (GENERATE) for the CX and sends it over the HTTPs session along with the user SALT. The SALT is used in concatenation with the PWD to create a key (PWD+SALT) to encrypt the AID$_{CX}$ before it is stored on the local browser storage. Server also stores the AID$_{CX}$ in its database (ADD$_{DB}$). Fig.2 shows the message exchanges.

C. Login Phase (LP)

In the login phase, the user enters the PWD on the CX, and the BT$_{ADDR}$ of the selected device connected to the PC is fetched. The BT$_{ADDR}$ and the encrypted AID$_{CX}$ (stored in the local browser storage) decrypted using the user PWD are sent to the website. The initial decryption of encrypted AID$_{CX}$ with user PWD thwart attacks via malwares which can steal encrypted AID$_{CX}$ from the memory. The website identifies the user based on the BT$_{ADDR.}$ The website encrypts the AID$_{CX}$ stored in its database using PWD + SALT and then decrypts it using the PWD. The decrypted value is matched with the value sent by the user. In the case of a match, it sends the SALT to CX to decrypt (D) the encrypted AID$_{CX}$ using the PWD + SALT. CX then uses the AID$_{CX}$ to encrypt the user's PWD and sends it to the website for user authentication. Also, a new AID$_{CX\_NEW}$ is created by the website during each login session and is shared with the CX.

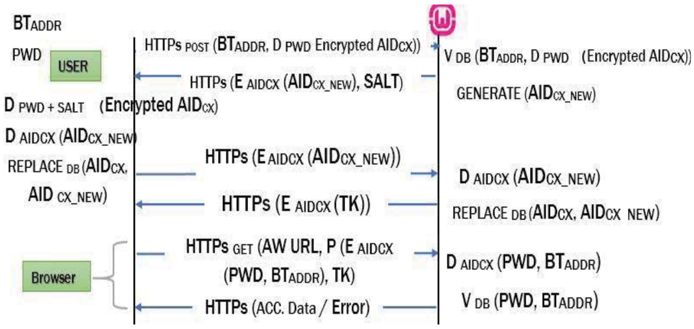

BT$_{ADDR}$
PWD
USER
HTTPs $_{POST}$ (BT$_{ADDR}$, D $_{PWD}$ Encrypted AID$_{CX}$))
V $_{DB}$ (BT$_{ADDR}$, D $_{PWD}$ (Encrypted AID$_{CX}$))
HTTPs (E $_{AIDCX}$ (AID$_{CX\_NEW}$), SALT)
GENERATE (AID$_{CX\_NEW}$)
D $_{PWD + SALT}$ (Encrypted AID$_{CX}$)
D $_{AIDCX}$ (AID$_{CX\_NEW}$)
REPLACE $_{DB}$ (AID$_{CX}$, AID $_{CX\_NEW}$)
HTTPs (E $_{AIDCX}$ (AID$_{CX\_NEW}$))
D $_{AIDCX}$ (AID$_{CX\_NEW}$)
HTTPs (E $_{AIDCX}$ (TK))
REPLACE $_{DB}$ (AID$_{CX}$, AID$_{CX}$ $_{NEW}$)
Browser
HTTPs $_{GET}$ (AW URL, P (E $_{AIDCX}$ (PWD, BT$_{ADDR}$), TK)
D $_{AIDCX}$ (PWD, BT$_{ADDR}$)
HTTPs (ACC. Data / Error)
V $_{DB}$ (PWD, BT$_{ADDR}$)

Figure 3. Login Phase

Figure 4. Intex FitRist

The CX replaces (REPLACE$_{DB}$) the existing AID in its local storage with the new AID$_{CX\_NEW}$ in encrypted form. The new AID$_{CX\_NEW}$ is used during the next login. However, the current AID$_{CX}$ is used to encrypt the ongoing communication of the current session. A session token (TK) is also shared by the server to identify the current session. The CX now constructs a HTTPs GET message containing the (Authentic Website) AW URL with parameters (encrypted with AID$_{CX}$) including BT$_{ADDR}$, PWD and TK (in plaintext). Once the server receives the HTTPs request from the user's browser, it extracts the token (TK) and identifies the user session to use the AID$_{CX}$ associated with the user to decrypt the parameters in the HTTPs GET request. The decrypted parameters are verified to allow user login. Fig.3 shows the message exchanges.

III. Implementation and Test Set Up

The scheme has been tested over the Chrome browser. The Web.Bluetooth API, which provides websites with access to the BT devices connected to a PC, was used to fetch the BT$_{ADDR}$ during WRP, ERP and LP. Testing was done with an Android OS supporting the Web.Bluetooth API. The website was hosted at HEROKU's server. An INTEX smart wearable named FITRIST (shown in Fig. 4) was used as the BT device for user identification.

As support for Bluetooth APIs to be used directly via a CX is currently not available, we used a CX to display an interface of the website login page so that the website opened on the CX could use the Web.Bluetooth API to complete the steps of BBAE authentication. The user can associate a CX with his Chrome browser login session. Hence, a CX provides user mobility as the storage associated with the CX (AID$_{CX}$) can be synced to another PC whenever the user logs in to his Chrome browser from that PC.

IV. Security Analysis

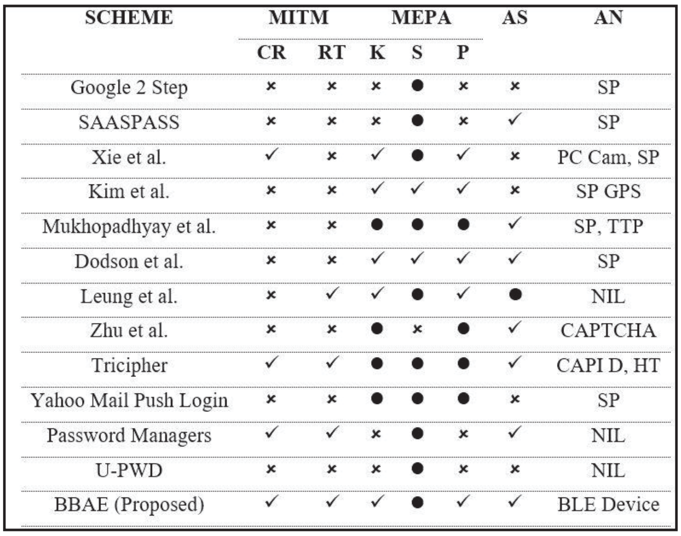
There are two main reasons for using BT$_{ADDR}$ as a user identification token: First, there is no need to enter it manually. Second, even if the attacker or a malicious insider sniffs the BT$_{ADDR}$ of the device, he cannot use it directly for authentication. This is because it is read by the CX (which has the AID$_{CX}$ needed for user authentication), and the attacker must spoof it on some BT device so that the CX can read it. The first level of security for AID$_{CX}$ is provided by the Chrome browser. The data stored in the CX storage cannot be accessed by peer APPs or CXs due to the cross origin and strong isolation properties. The second level of security is provided by encrypting the AID$_{CX}$ with the user's PWD and SALT before storing it in the CX storage. We have compared BBAE with existing schemes in terms of known security threats including MITM (RT/CR), ME based Phishing Attacks (MEPA) (Key logging: K, Screen logging: S, Password Sniffing: P), App Spoofing (AS) (where spoofed SP/Browser apps or extensions can be installed covertly to steal user credentials to compromise an authentication scheme), and Additional Needs (AN). Fig.5 shows the comparison (✓: Safe from attack, ✗: Unsafe from attack, ●: Safe from attack but partial credentials can be compromised). The RT/CR MITM phishing, MEPA or the app/extension spoofing attacks cannot get access to all the credentials needed for authentication in the case of BBAE.

We also did a survey to identify if people would use authentication schemes which utilize Bluetooth such as BBAA in the near future. 100 people participated in the online survey at surveymonkey.com. 70 of these people identified themselves as being computer proficient, 28 stated they were not but use a computer frequently, and 2 stated that they use a computer infrequently. We explained the working of the schemes including: **(1)** UNAME-PWD (*S1*), **(2)** UNAME-PWD-OTP (*S2*) **(3)** QR Code (*S3*) **(4)** Graphical PWD (*S4*) **(5)** Hardware/Security token (*S5*), **(6)** BBAE, and **(7)** Push login based (*S7*). Two of the important questions included in the survey were:

(A) Which scheme would they prefer to use while logging in to banking or important web accounts, and

(B) Which scheme they thought was easy to learn and use while also being secure, and which they want to see in future implementations on the website.

The obtained results (in terms of the number of people who voted for a scheme) are given in Fig. 6. The number of votes received from the users reveals that BBAE was trusted more than S1, S3 and S4 schemes for logging into important web accounts and was voted as a preferred scheme for future over S1, S3, S4 and S5. Even when users had not used BBAE or similar types of schemes (this fact was learned from a verbal discussion with the participants), the obtained statistics and confidence in BBAE as compared with other schemes shows that BBAE could be a great success in the future.



SCHEME	MITM		MEPA			AS	AN
	CR	RT	K	S	P		
Google 2 Step	×	×	×	●	×	×	SP
SAASPASS	×	×	×	●	×	✓	SP
Xie et al.	✓	×	✓	●	✓	×	PC Cam, SP
Kim et al.	×	×	✓	✓	✓	×	SP GPS
Mukhopadhyay et al.	×	×	●	●	●	✓	SP, TTP
Dodson et al.	×	×	✓	✓	✓	✓	SP
Leung et al.	×	✓	✓	●	✓	●	NIL
Zhu et al.	×	×	●	×	●	✓	CAPTCHA
Tricipher	✓	✓	●	●	●	✓	CAPI D, HT
Yahoo Mail Push Login	×	×	●	●	●	×	SP
Password Managers	✓	✓	×	●	×	✓	NIL
U-PWD	×	×	×	●	×	×	NIL
BBAE (Proposed)	✓	✓	✓	●	✓	✓	BLE Device

Figure 5. Comparison of BBAE with existing schemes

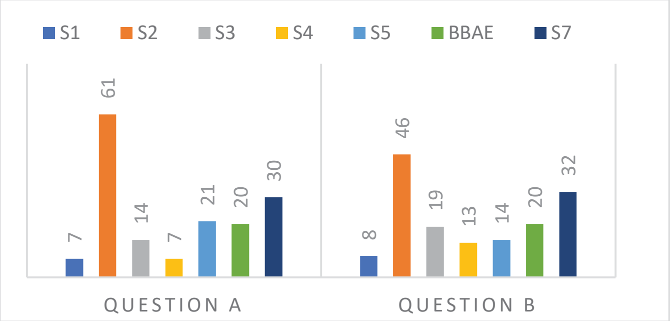


Figure 6. User survey statistics

V. Conclusions

The proposed scheme can handle RT/CR MITM, MEPA, and other sophisticated credential stealing attacks. The novelty of the scheme is that it uses BLE/BT smart devices as a secure token for user identification instead of using it as an additional token (as done in the SAASPASS scheme) or as a communication channel for transferring user authentication information (as done in Czeskis et al. [17] and Yubico [13]). In addition, unlike SAASPASS, it does not require users to install a client side operating system dependent module and secures users from attacks that can occur via BT address spoofing.

VI. Appendix

- RT MITM to hack a user Gmail login using a phishing website. https://youtu.be/nccDJMIkJtc
- RT MITM Phishing attack on SAASPASS secret PIN based authentication. https://youtu.be/y-w0RfCaBrQ
- Malicious extension to sniff the username and passwords entered over the websites by Lastpass password manager. https://youtu.be/uNwahzlzpSg .
- QR code relaying over email to compromise WhatsApp web authentication. https://youtu.be/6FThk1Iystw
- A video demonstrating key logging and phishing (Browshing) via MEs. It also demonstrates that accessing user information and keystrokes on peer apps is not possible due to cross origin property. **https://youtu.be/fLfzsAodsBw**

References

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

[2] 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, 232-239, 2015.

[3] 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, 51-62, 2013.

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

[5] G. Varshney, M. Misra, and P. Atrey, "Browshing - a new way of phishing using a malicious browser extension," presented at the International Conference on Innovations in Power and Advanced Computing Technologies (IPACT 17) (In Press), Vellore, Chennai, India, 2017.

[6] L. Liu, X. Zhang, G. Yan, and S. Chen, "Chrome Extensions: Threat Analysis and Countermeasures," in NDSS, 2012.

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

[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. 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, 505-508, 2011.

[11] 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, 17-38, 2010.

[12] 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, 891-904, 2014.

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

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

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

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

[17] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz, "Strengthening user authentication through opportunistic cryptographic identity assertions," in Proceedings of the 2012 ACM conference on Computer and communications security, 404-414, 2012.