RESEARCH PAPER | Engineering | Volume : 5 | Issue : 3 | March 2015 | ISSN - 2249-555X

QR-Based OTP Authentication to Prevent Keylogging

KEYWORDS: Keylogging, phishing, pharming, session hijacking, QR code, authentication, malicious code, attack, android, visualization

R.Divya
Department of CSE, S.A. Engineering College, Tamil Nadu, India.

T.Sennila
Department of CSE, S.A. Engineering College, Tamil Nadu, India.

ABSTRACT *Keylogging is an activity of capturing users' keyboard strokes and records the activity of a computer user in a covert manner using keylogger hardware and software. The keyloggers secretly monitors and log all keystrokes. Unlike other malicious programs, keyloggers do not cause any threat to system. But it can be used to intercept passwords and other confidential information entered via the keyboard by considering various rootkits residing in PCs (Personnel Computers) that breaches the security. Cyber criminals can get user names, email passwords, PIN codes, account numbers, email addresses, passwords to online gaming accounts, e-payment systems, etc. As a result, it impersonates a user during authentication in financial transactions. To prevent keylogging, the strict authentication is required. The QR code can be used to design the visual authentication protocols to achieve high usability and security. The two authentication protocols are Time based One-Time-Password protocol and Password-based authentication protocol. Through accurate analysis, the protocols are proved to be robust to several authentication attacks. And also by deploying these two protocols in real-world applications especially in online transactions, the strict security requirements can be satisfied.*

INTRODUCTION

Keyloggers are used as an surveillance tool by the employers to ensure employees use work computers for business purposes only. Unfortunately, keyloggers are embedded in spyware and allows the information to be sent to unkown third party. Keyloggers can be used in some IT organizations to troubleshoot technical problems in computers and business networks. Keyloggers are also used by a family or business to monitor the people without their knowledge. Finally, keyloggers are installed in public kiosks to steal credit card information or passwords.

Keylogging allows malicious software to capture keyboard strokes whenever the user types in the specific application or forms to obtain the passwords. Cyber criminals use keylogging to capture credentials and authentic information to hack the account and performs financial fraudulence and therefore gains access to confidential information. Malware uses several techniques to log keystrokes such as hooking into the keyboard driver and other operating system services. The keylogger is present both in personal and public computers and it is pervasive. Keyloggers are often root kitted. So the presence of keyloggers cannot be detected.

To overcome keylogging attack, virtual or onscreen keyboards are used. Both the techniques rearrange the alphabets randomly and therefore frustrate simple keyloggers. But the keylogger has control over the entire PC, which can capture every event and read the video buffer to create a mapping between the clicks and new alphabet.

The keylogging attack is quite similar to the shoulder-surfing attack where the attacker sees the direct input of the client to the computer and also every behavior of the client. Many graphical password schemes are introduced to prevent shoulder-surfing attack. But many of these schemes are unusable. Even though the cryptographic secrets are securely delivered to the client's PC, the attacker residing on the client's PC can easily observe and deceive the information.

To solve this problem, the intermediate device between human and terminal is introduced. This helps to design a human involving protocol. Every interaction between the client and the intermediate device is visualized using Quick Response (QR) code. In these protocols, the client does not need to memorize any information other than password and PIN. However, the authentication process can be visualized which enhances security and usability to the client. The security protocol has the client involvement using smartphone with augmented reality. A smartphone with camera is used to visualize the authentication process.

Instead of implementing the entire security protocol in computer, a part of it is moved to the smartphone. This visualization in smartphone offers protection against malware, keylogging attacks and shoulder-surfing attacks.

SYSTEM MODEL

The system model comprises of four different entities such as a client, a smartphone, a client's terminal (PC) and a server. The client is a user or an ordinary human with limited capabilities of remembering cryptographic credentials such as keys and performing complex mathematical computations. A client's terminal is a client's PC which is used to connect to a server for performing financial transactions. The client has the smartphone which stores the public key certificate of the server or digital certificate equipped with a camera. The server is the system entity belongs to the financial institution which interacts with the user by performing all the back-end operations.

The client or an user is registered in a particular bank for performing online transactions and provided with the unique client ID and password. The registered client can log on to particular bank site. The client must enter into retail login. When the client sends the unique ID to the server, the server checks the client's information from the bank database. If the client's information is correct, the server retrieves the public key and fresh random time-based one-time-password (TOTP) from the database.

Volume : 5 | Issue : 3 | March 2015 | ISSN - 2249-555X

The server generates the QR code which comprises of unique client ID, public key, TOTP and time slot. Then the QR code is sent to the client. On client's terminal, the QR code is displayed. Now, the client has to take his smartphone in which the QR code scanning application is already installed. The QR code has to be scanned. After scanning the QR code, the decoded information will be displayed in the smartphone. The randomized keyboard which looks like a 4x4 matrix with random arrangements of 0-9 digits is displayed in the smartphone.

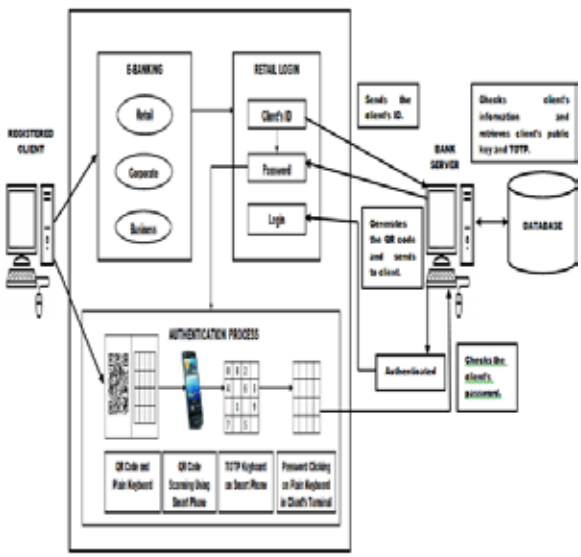

Figure 1: shows the overall system architecture. Here, the e-banking is taken as an example to show how the authentication process works.

On the client's terminal the password box is replaced with the 4x4 blank keyboard matrix. Now, the client has to just click on the rows or columns of the blank keyboard matrix by seeing where is password has been arranged in the smartphone. From the client's terminal, only the ID of the keyboard matrix is sent to the server. The server also does not know the password of the client. Based on the ID of the keyboard matrix, the client gets authenticated. If the client clicks on the wrong ID, again the previous steps are repeated by sending a newly generated QR code to the client. And also if the client fails to login within the allotted time slot, the server will automatically generates a new QR code with new TOTP. After the client gets authenticated, the client can enjoy all the e-banking services.

TIME-BASED ONE-TIME-PASSWORD PROTOCOL

In this section, a Time-based One-time-password authentication protocol is introduced which is referred as first protocol. It makes use of random string for authentication. This protocol works as follows:

- The client sends the unique client ID to the server.
- The server checks the client's information from the database and retrieves the client's public key (PK_{ID}).
- The server then picks a fresh random string TOTP with a time slot and encrypts it with the public key to obtain

E_{TOTP} = Encr(PK_{ID} (TOTP)). Eq. no. (1)

The server generates the QR code and sends it to the client.

- In the client's terminal, a QR code QR_{EOTP} is displayed.
- The client decodes the QR code with

E_{TOTP} = QRDec(QR(E_{TOTP})).Eq. no. (2)

- The random string is encrypted with client's public key (PK_{ID}), the client can read the TOTP string only through her smartphone by

TOTP = Decrk(E_{TOTP}) Eq. no. (3)

and type in the TOTP in the terminal with a physical keyboard.

- The client has to type the TOTP in the terminal where the keyboard matrix is displayed.
- The server checks the result entered by the client and if it matches what the server has sent earlier, the client is authenticated.

If the client does not authenticated, the access is denied.

PASSWORD-BASED AUTHENTICATION PROTOCOL WITH RANDOMIZED ONSCREEN KEYBOARD

In this section, the second protocol password-based authentication protocol is described. Here, the password is shared between server and client, and a randomized keyboard. The protocol works as follows:

- The client connects sends unique client ID to the server.
- The server checks the received unique client ID to retrieve the client's public key (PK_{ID}) from the database.
- The server prepares a random permutation of a keyboard arrangement, and encrypts it with the public key to obtain

E_{KBD} = Encr ($PK_{ID}(\pi)$). Eq. no-(1)

- The server encodes the ciphertext with QR encoder to obtain
- QR(E_{KBD}) = QR(Enc(E_{kID} (π))). Eq. no-(2)

- The server sends the result to the client with a blank keyboard.
- In the client's terminal, a QR code (QR(E_{KBD})) is displayed together with a blank keyboard.
- The onscreen keyboard does not have any alphabet on it, the client cannot input her password.
- The client executes her smartphone application which first decodes the QR code by applying

QR(Dec(QR(E_{KBD})) Eq. no-(3)

to get the ciphertext (E_{KBD}).

- The ciphertext is then decrypted by the smartphone application with the private key of the client to display the result on the smartphone's screen

π=Decr(SKID(EKBD)). Eq. no-(4)

- When the client sees the blank keyboard with the QR code through an application on the smartphone that has a private key, alphanumerics appear on the blank keyboard and the client can click the proper button for the password.
- The client types in her password on the terminal's screen while seeing the keyboard layout through the smartphone.

Volume : 5 | Issue : 3 | March 2015 | ISSN - 2249-555X

- The terminal does not know what the password is but only knows which buttons are clicked.
- Identities of the buttons clicked by the client are sent to the server by the terminal.
- The server checks whether the password is correct or not by confirming if the correct buttons have been clicked.

TABLE 1

Comparison of two protocols and their resistance to different attacks when the terminal and the smartphone are under control of the attacker.

Protocol	Attack	Brute-force	Keylogger	Malware
TOTP	Smartphone	Yes	Yes	No
	Terminal	Yes	Yes	Yes
PAP	Smartphone	Yes	Yes	Yes
	Terminal	Yes	Yes	Yes

CONCLUSION

The two authentication protocols are proposed to show how visualization can enhance usability and security. Moreover, these two protocols helps to overcome many challenging attacks such as keylogging and other malware attacks. This system can be implemented in many real world applications since it utilizes simple technologies and feasible to use as android application.

REFERENCE [1] BS ISO/IEC 18004:2006. Information Technology. Automatic Identification and Data Capture Techniques. ISO/IEC, 2006. | [2] C.-H. O. Chen, C.-W. Chen, C. Kuo, Y.-H. Lai, J. M. McCune, A. Studer, A. Perrig, B.-Y. Yang, and T.-C. Wu. Gangs: gather, authenticate 'n group securely. In J. J. Garcia-Luna-Aceves, R. Sivakumar, and P. Steenkiste, editors, MOBICOM, pages 92–103. ACM, 2008. | [3] M. Farb, M. Burman, G. Chandok, J. McCune, and A. Perrig. Safeslinger: An easy-to-use and secure approach for human trust establishment. Technical report, CMU, 2011. | [4] H. Gao, X. Guo, X. Chen, L. Wang, and X. Liu. Yagp: Yet another graphical password strategy. In Proc. of ACM ACSAC, pages 121–129, 2008. | [5] E. Hayashi, R. Dhamija, N. Christin, and A. Perrig. Use your illusion: secure authentication usable anywhere. In Proc. of ACM SOUPS, 2008. | [6] A. Hiltgen, T. Kramp, and T. Weigold. Secure internet banking authentication. IEEE Security and Privacy, 4:21–29, March 2006. | [7] J. Katz and Y. Lindell. Introduction to modern cryptography. CRC Press, 2008. | [8] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder surfing by using gaze-based password entry. In Proc. of ACM SOUPS, pages 13–19, 2007. | [9] Y.-H. Lin, A. Studer, Y.-H. Chen, H.-C. Hsiao, E. L.-H. Kuo, J. M. McCune, K.-H. Wang, M. N. Krohn, A. Perrig, B.-Y. Yang, H.-M. Sun, P.-L. Lin, and J. Lee. Spate: Small-group pki-less authenticated trust establishment. IEEE Trans. Mob. Comput., 9(12):1666–1681, 2010. J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In Proc. of IEEE Symposium on Security and Privacy, pages 110–124, 2005.