International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 1, January- 2013

Cryptographic Provenance Verification For Secure Hosts

P .Angel
II M.E, Computer Science and Engineering,
Sri Shakthi Institute of Engineering and
Technology, Coimbatore, India.

S.Hemalatha
Assistant Professor, Department of Computer
Science and Engineering, Sri Shakthi Institute of
Engineering and Technology, Coimbatore, India.

Abstract

To improve the trustworthiness of a host and system data cryptographic verification technique is used. From these approach users keystroke to identify the legitimate user's to get the service from the server by using integrity verification and malicious traffic detection. But the main issue of this approach is keystroke hacking by the bots or autonomous program. The key logger's records the user's password or security code and send to the hacker to login the service when the user is no longer online. This project aims to propose a malware detection approach based on the characteristic behaviour of human user's. Finally explore the human-bots differences by using TUBA (Telling hUman and Bots Apart) thereby prevent bots forgeries. By using classifier algorithm the current work identify the human and malware activities.

Keywords— Malware, TUBA, Cryptography, Classifier.

1. Introduction

Authentication and Identity theft remains one of the most prevalent issues on the Internet today. To authenticate the legitimate user's fall into three categories, they are Ownership factor such as wrist band, ID card, security token, knowledge factor such as a password, phrase, and inherence factor such as fingerprint, retinal pattern, signature, face, voice, and other biometric identifier. Each factor is used to authenticate a person's identity in prior to access the restricted services, sensitive data, a transaction request, User authentication to identify the user and verify that the user is allowed to access some the services that are restricted. When the user log in to the network account to verify that the users are authorized to use the restricted computing resources, and, additionally, it shows they are the owner of the particular set of those resources like files, e-mail, and so on by giving the correct user id and password.

Identity theft is handled by the malware.Malware, is also called as malicious (or malevolent) software, this software used or created by attackers to disrupt system operations, collect and transmit the sensitive information to others. It can appear either in the form of code, scripts or active content. Simply it is a software program that does many malicious things. Malware entities stealthily residing on a user's computer and interacting with the user's Computing resources without the user permission.

Key loggers are one type of malware which record every key stroke that is pressed by the users of a computer. They can perform many illegal activities like taking screenshots, monitoring the system resources and user activities.

To authenticate the users identity and protect the user's data from key loggers. We define new security property is data provenance integrity. It's used to improve the trustworthiness of a host and the system data. There are two approaches for ensuring the system properties one is keystroke integrity verification another one is malicious traffic detection.

Cryptographic mechanisms is used to ensure the correct data flow, system properties of a host, and secure transaction or communication between client and server, especially on verifying the dynamic system-related data's provenance. This paper illustrates how to sign and verify the user's keystroke events that are from external keyboard devices between client-server architecture, i.e., verifying the provenance of user's keystrokes in net banking application.

1.1 Our contributions:

This paper presents new cryptographic provenance verification for realizing user's keystroke integrity and host based traffic monitoring.

- We propose the security models and operation in cryptographic provenance verification. It ensure the data flow in hosts, system related data and the properties.
- Construction of light weight cryptographic protocol. It's used to prevent malicious bots from injecting keystroke events into a host's application by generating cryptographic keys.
- Cryptographic provenance verification approach is applied for realizing a host based monitoring framework.

2. Related Work

Our paper focuses on network based authentication approach for ensuring user's keystroke integrity and also protect our sensitive information from key loggers(malware).Here we develop these works for net banking application. Initial step to utilize the net banking resource the users must have the bank account while creation the user's personal details are stored. The authentication and validations are carried out at the bank server manually.

After bank account creation the user get username and password for utilize the net banking application. The user successfully login the first time it requested to change the account setting that include the password. While changing the password the corresponding user's keystrokes events such as flight time, dwell time are captured and key count is stored in to the bank server data base. When the user login, the application each time it will check the key count along with the signature. If the user login within the key count as well as their signature is verified successful means the user is considered as a legitimate and they can get the service from the server if not verified means it mark it as a imposter and do not allow the user to utilize the net banking application.

The server can terminate the user service when the malicious activities like key loggers present in the system. The key loggers are identified by the server when it monitors the changes in IP address of the user.

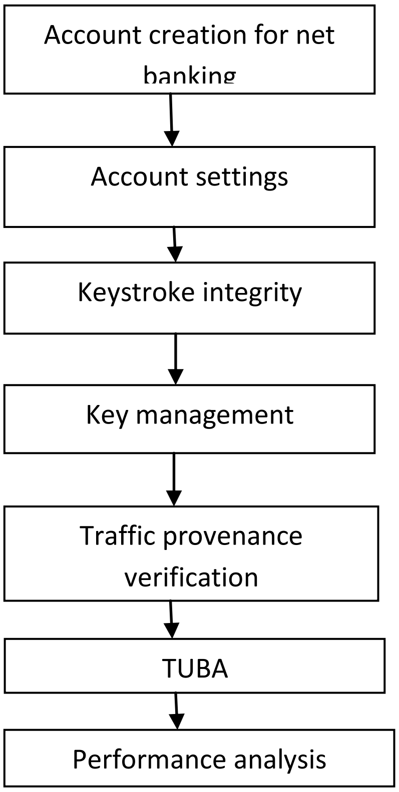

Fig1.flow diagram

3. Cryptographic Provenance Verification

CPV mechanism is used to ensure the true origin of the data such as system device or legitimate user. It can be implemented by generating digital signature .they can be signed at the source itself and are verified at the server side. In CPV approach signer and verifier is a program instead of person. The main objective is to prevent unauthorized use of net banking application by a key loggers and malicious bots. Our goal is mainly focused to answer this question "this application is being used by the authenticated owner i.e. legitimate users or by an imposter?"To avoid spoofing the content, injecting forgeries and tampering by the malware in between client and the bank cryptographic signature scheme is used.

In CPV there are three major operations are carried out .they are setup, keystroke sign and keystroke verify. In setup the signing key and the bank server sets the verification key as their own public key. In sign phase whatever the client generates the data can be signs by using the client private key and send along with the signature. At verification phase the bank server receives the signature and check the origin of the source whether it comes from the legitimate one by verify their signature by using server's public key. Digital signatures are generated by using UMAC (message authentication code using Universal hashing).

4. Architecture of Keystroke Integrity

It establishes a secure channel between user's host and the remote bank server. Each message contains an encrypted form of user's keystrokes and its signature. This module mainly deals with key code, flight time and dwell time. Key code is the ASCII code that represents each key in a keyboard. Flight time is also called as Down-Down Time. DD time is a keystroke latency defined as the time interval between each successive keystroke. Dwell time is also called as Up-Down Time that is time taken for key pressed and key released. When the customer changes the password the key strokes are captured and key count is calculated. Finally the key count is stored in the database as an integer for check the keystroke integrity of the user. In timing Accuracy the basic foundation for the typing biometrics is to have an accurate and reliable data source of typing patterns in time. In this paper, the time stamp counter function was used to catch the count of clock cycles. The time stamp counter keeps an accurate count of every cycle that occurs in the system.

Each time the user tries to log on the net banking application he must type the account number, user name and target string like password. While the user is typing, keystroke data's are captured and they analysed if the sample is a account's owner. If the sample is considered true, then the user could access the application otherwise it mark it as an illegitimate user.

- **Key Code**: Key code is the ASCII code that represents each keys present in a keyboard. When a string contains capital letters, there are more than one possible set of key codes, otherwise there is a single one.Ca = fc1(a); c2(a); . . . ; cn(a)g denotes the key codes contained in the template of the account a and Ca;w = fc1(a;w); c2(a;w); . . . ; cn(a;w)g denotes the key codes contained in the sample w in the account a.
- **Down-Down Time:** DDtime is a keystroke latency defined as the time interval between successive keys. This feature is represented by DDa;w = fdd1(a;w); dd2(a;w); . . . ; ddn(a;w)g, where ddi(a;w) = ti+1_down(a; w) - ti_down(a;w) is relatedto (ki; ki+1).
- **Up-Down Time*:*** UD time is also a keystroke latency feature representedby UDa;w = fud1(a;w); ud2(a;w); . . . ; udn+1(a;w)g, where udi(a;w) = ti+1_down(a;w) -ti_up(a;w) is related to (ki; ki+1). the UD separation between the true samples and the false samples. This feature Could be positive or negative according to two situations. In the first situation, ki+1 is only pressed when ki was released which results in a positive time value. In a second situation, ki+1 is pressed while ki is also pressed, which results may show in a negative time value.

4.1Key Management in Keystroke Integrity

Key management develops the keys for the user key count. The keys are used to protect the user's key count from the malware. While password is changed the trust agent generates two pseudorandom numbers (a_0, a_1) for the user and sign by the sign key and the generated data are encrypted using the bank server's public key. Then it generates two random numbers (b_0, b_1) and encrypts them using the public key of the trust agent. Bank Server and the trusted agent exchange the encrypted random numbers and XOR the decrypted values with the sent bits to use as two symmetric keys using one key for signing, and another key for encrypting the user's data. Finally, the server verifies the digital signature. When the trust agent disconnects, the binding key is used to bind the symmetric keys for bank server and the user and it can securely store them so the key exchange is not required during the next connections.

4.2Tracking Provenance for Outbound Service

In this section, it illustrates the cryptographic provenance verification approach in a networking based application, in particular for ensuring the integrity of outbound packets, it describe how the user request are flow to the remote server This module deals with the secure communication between client and server. RSA algorithm is used for encryption and decryption purpose. The bank user initiates the connection with the server by login the net banking. The server authenticate the user by checking the key count of the user If the keystrokes are successfully verified means it allow the user's to get the service from the server. The client and server exchange their public keys. The user generates two random numbers a0 and a1, and encrypts a0 and a1 using the servers public key. The user sends encrypted a0 and a1 to the server. Then the server Receives and decrypts a0 and a1 with its private key. It then generates two random numbers b0 and b1. The server encrypts b0 and b1 using the user's public key. Customer decrypts them with its private key. Both the client and server modules have a0, a1, b0, and b1. They compute the signing key as a0 + b0 and the symmetric key for their communication encryption as a1 + b1.If the customer keystrokes timing and their signature is successfully authenticated by the bank server then the customer successfully login and they can able to use the application.

In Password selection is the first potential Confounding factor we identified. Some passwords can be typed faster than others. The choice of password may affect a subject's keystroke times, distorting the effect of clock resolution. To control for the potential confounding factor, we chose a single fixed but representative password to use throughout the experiment.

The data-collection application was installed on a Single laptop with no network connection and with an external Keyboard. We identified keyboard selection as another Potential confounding factor. If subjects used in different keyboards and the difference might affect their keystroke times. We control for the potential confounding factor by using one keyboard throughout the experiment. Telling hUman or Bot Apart (TUBA) is used to differentiate whether the application is currently used by a legitimate human or malicious Bot.

4.3Performance Analysis

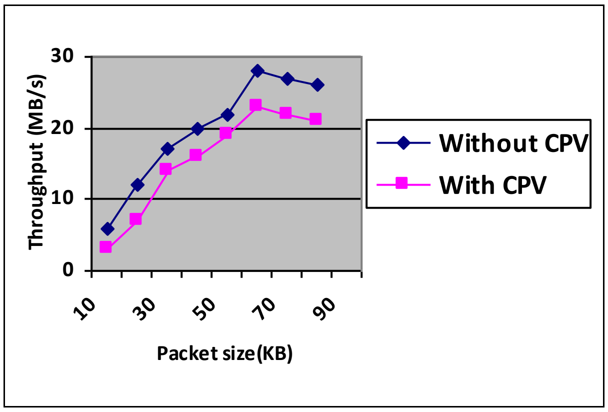

Fig.2.Comparison on outbound packet with or without signs

When we apply a CPV approach to the users keystrokes Fig.2 illustrates that the throughput is slightly degrade and the security is too high compare to the data without using provenance approach .While increasing the security the cost of signing and verification of data also amortized.

5. Conclusions

The main objective of this paper is to improving the assurance of system data and properties by preventing and identifying malware activities. The system security solutions against malware complement and network-traffic-based analysis.

This project demonstrated CPV to distinguish legitimate and illegitimate user based on the user's keystrokes. it authenticates the user's keystrokes in static manner. The technical contributions in this project: 1) it proposed the model and operations of cryptographic provenance verification of user's keystrokes in a network-based security setting. 2) It demonstrated the data provenance verification approach in a lightweight cryptographic framework for ensuring the integrity of outbound packets or outbound request from a host. This traffic-monitoring or communication monitoring framework creates checkpoints that cannot able to bypass by the malware activities or malware traffic.3) An efficient keystroke integrity verification protocol in a client and net bank server architecture that prevents malware from forging keystroke events. This keystroke-integrity service serves as an important building block for the future construction of human behaviour- driven security solutions by using classifier and providing host based malware identification.

References

[1] A. Baliga, V. Ganapathy, and L. Iftode, "Automatic Inference and Enforcement of Kernel Data Structure Invariants," Proc. 24th Ann.Computer Security Applications Conf. (ACSAC '08), 2008.

[2] A. Baliga, P. Kamat, and L. Iftode, "Lurking in the Shadows: Identifying Systemic Threats to Kernel Data," Proc. IEEE Symp.Security and Privacy, pp. 246-251, 2007.

[3] B. Blackburn and R. Ranger, Barbara Blackburn, the World's Fastest Typist. 1999.

[4] M. Christodorescu, S. Jha, and C. Kruegel, "Mining Specifications of Malicious Behavior," Proc. Sixth Joint Meeting of the European Software Eng. Conf. and the ACM SIGSOFT Symp. the Foundations of Software Eng. (ESEC-FSE '07), pp. 5-14, 2007.

[5] W. Cui, R.H. Katz, and W. tian Tan, "Design and Implementation of an Extrusion-Based Break-in etector for Personal Computers," Proc. 21st Ann. IEEE Computer Security Applications Conf. (ACSAC '05), pp. 361-370, 2005.

[6] D.E. Denning, "A Lattice Model of Secure Information Flow," Comm. ACM, vol. 19, pp. 236- 243,May 1976.

[7] D.E. Denning and P.J. Denning, "Certification of Programs for Secure Information Flow," Comm. ACM, vol. 20, pp. 504-513, July 1977.

[8] M. Dhawan and V. Ganapathy, "Analyzing Information Flow in Javascript-Based Browser Extensions," Proc. Ann. IEEE Computer Security Applications Conf. (ACSAC '09), pp. 382-391, 2009.